



Filtrage en ligne des évènements sur une expérience de production de dimuons de haute masse

Jean-Pierre Froberger

► To cite this version:

Jean-Pierre Froberger. Filtrage en ligne des évènements sur une expérience de production de dimuons de haute masse. Physique des Hautes Energies - Expérience [hep-ex]. Université Louis Pasteur - Strasbourg I, 1980. Français. NNT: . tel-00396173

HAL Id: tel-00396173

<https://theses.hal.science/tel-00396173>

Submitted on 17 Jun 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



14 OCT. 1980

EX - CRN - HE 80-11

C. R. N.

herches nucléaires de Strasbourg

CPN/HE 80-11

THESE

présentée

pour obtenir le grade de

DOCTEUR DE SPÉCIALITE

Physique Nucleaire et Corpusculaire

par

Jean Pierre FROBERGER

FILTRAGE EN LIGNE DES EVENEMENTS SUR UNE EXPERIENCE
DE PRODUCTION DE DIMUONS DE HAUTE MASSE.

CERN LIBRARIES, GENEVA



CM-P00050484

Thesis-1980-Froberger

Institut National
de Physique Nucléaire
et de Physique
des Particules

Université
Louis Pasteur
de Strasbourg

THESE

CRN/HE 80-11

présentée

A l' U.E.R. DES SCIENCES DE LA MATIERE
DE L'UNIVERSITE LOUIS PASTEUR DE STRASBOURG

pour obtenir le grade de

DOCTEUR DE SPÉCIALITE

Physique Nucleaire et Corpusculaire

par

Jean Pierre FROBERGER

FILTRAGE EN LIGNE DES EVENEMENTS SUR UNE EXPERIENCE
DE PRODUCTION DE DIMUONS DE HAUTE MASSE.

Soutenue le 1 Juillet 1980 devant la Commission d'Examen:

MM. G. MONSONEGO

J. M. MEYER

J. LECOQ

J. J. BLAISING

President

} *Examineurs*

U.E.R. DES SCIENCES EXACTES

Mathématiques

Sciences Physiques et Chimiques

Sciences de la Vie et de la Terre

Sciences du Comportement et de l'Environnement

Ecole d'Application des Hauts Polymères

Observatoire

Physique du Globe

Xavier FERNIQUE (Par intérim)

Henri BENOIT

Jean José FRIED

Yves BOULLANGER

Bruno WILL

Constant WIPPLER

Marc DAIRE

Alphonse FLORSCH

Roland SCHLICH

Directeur

Directeur

Directeur

Directeur

Directeur

Directeur

Directeur

DOYENS HONORAIRES : P. LACROUTE - J.H. VIVIEN - G. MILLOT

PROFESSEURS HONORAIRES : J. BRENET - J. BYE - H. CARTAN - C. CHABAUTY - A. CHRETIEN - J. DENY - M. S. GILLET - S. GORODETZKY - R. HOCART - P. JOLY - P. LACROUTE - R. LECOZET - G. LEMSE - P. L'HERITIER - J. KERNEROWICZ - A. MAILLARD - L. NEEL - J. PARROT - R. ROHMER - J.P. ROTHE - L. SACKMANN - Ch. SADRON - H. SAUCIER - F. STUTINSKY - H. VILLAT - Et. WOLFF

MAITRE DE CONFERENCES HONORAIRE : R. WEIL

PROFESSEURS

J.P. ADOLFF	Chimie Nucléaire	J. FARAUT	Mathématiques	M. MIGNOTTE	Informatique
R. ARBERLSTER	Physique	P. FEDERLIN	Chimie	G. MILLOT	Géol. et Paléontologie
V. AVANISSIAN	Mathématiques	P. FERRIER	Physiol. Animale	G. MONSIEGO	Physique théorique
G. BARBANCON	Mathématiques	X. FERNIQUE	Mathématiques	G. OURISSON	Mathématiques 1
F. BECKER	Phys. Mathémat.	J.G. FISCHER	Mathématiques	J.P. RAMIS	Mathématiques
N. BEFORT (dett)	Bochimie	D. FOATA	Zoologie	G. REEB	Topologie
H. BENOIT	Dermato-Chimie	E. FOLLENIUS	Méc. des Fluides	R. RICHARD	Physiol. animale
H. BENOIT	Physicochimie Macrom.	J. FRIED	Math. Gén. Chim. Phys.	J. RIEHL	Chimie
H. BENOIT	Physicochimie Macrom.	J. FRIED	Math. Gén. Chim. Phys.	A. RICHE	Physique
D. BERNARD	Math. Math. de la Phys.	A. FUCHS	Botanique	A. ROCHET	Physicochimie
J.C. BERNIER	Chimie Générale	A. GAGNIEU	Botanique	R. ROUART	Physique
J. BONNIN	Géoph. interne	J.C. GALL	Géologie	F. ROUX	Physique
J. BONNIN	Chimie biologique	A. GALLMANN	Physique	F. SCHALLER	Biologie générale
J.F. BOUTOT	Mathématiques	F. GAUTIER	Mathématiques	G. SCHMITT	Mathématiques
V. BRINI	Chimie	G. GLASSER	Mathématiques	J. SCHWAB	Chimie
J. BROSSAS	Chimie macromol.	C. GLASSER	Mathématiques	M. SCHWAB	Physique
C. BURGGRAB	Minéralogie	C. GODBILLON	Mathématiques	M. SIEKINDIE	Chimie
H. BURNAGE	Mécan. des Fluides	M. GOUNOT	Physique	G. SOLLADIE	Chimie
G. CERF TIER	Physique générale	M. GROSSMANN	Physique	J. SOMMER	Chimie
F. CHEVALIER	Chimie	L. HIRTH	Chimie	G. SUTTER	Chimie
A. CLAUS	Chimie	J.P. JOULEAU	Microbiologie	G. TESSIER	Chimie
A. COCHE	Physique nucléaire	T. JOUEAU	Astronomie	J. THIÉBOLD	Chimie
M. DAIRE	Chimie	R. KIRSCH	Mathématiques	D. VIAUD	Chimie
H. DANAN	Phys. Atomique	F. LACROUTE	Mathématiques	J.H. VIVIER	Zool. et Embryol. Expér.
E. DANIEL	Phys. expérimentale	G. LAFON	Mathématiques	R. VOLTZ	Physique théorique
X. DELAND	Biophysique	G. LEBEURIER	Mathématiques	G. WEILL	Physique
A. DELUZARCHE	Chimie	M. LEROY	Mathématiques	R. WESS	Chimie
G. DUNOYER de	Chimie	J. LUCAS	Mathématiques	B. WILLY	Chimie
H. DURANTON	Géologie	D. MAGNAC	Mathématiques	C. WIPPLER	Physicochimie
J. BEL	Chim. Biologique	J. MARTINET	Mathématiques	J. WUCHER	Physicochimie des Hts Polym.
J.P. EBERHART	Minéralogie	P. MIALHE	Physiol. animale		
V. ERN	Physique	A. MICHARD	Géologie		

PROFESSEUR ADJOINT : J. SITTILLER (Géologie)

PROFESSEURS ASSOCIES :

A. BANDERET	E.A.H.P.	R. DUCKETT	Chimie EAHP	J. OSBORN	Chimie minérale
B. BOURQUILH	Géophysique	R. HOLMES	Chimie	C. SAVOY	Physique
G. BUCHANAN	Chimie	J. HONNOREZ	Géologie	Y. SIBUYA	Mathématiques
M. BURTICK	Physique	P. MARGARETHA	Chimie	L. WILLIAMS	Chimie
M. CONSTANTIN	Chimie	T. MIZOGUCHI	Physique	S. WILLIAMSON	Mathématiques

PROFESSEURS CONVENTIONNES : P. BOUVEROT (Physiologie respiratoire) - P. DEJOURS (Physiologie respiratoire)

ASTRONOME ADJOINT : A. FLORSCH (Astronomie)

DIRECTEURS DE RECHERCHE C.N.R.S. :

J.F. BELLMANN	Chimie	J. MEYER	Botanique
P. BOUTOT	Physiologie respiratoire	A. PORTE	Biologie cellulaire
P. DEJOURS	Physiologie	R. REMPP	Physicochimie macromoléculaire
A. KNIPPER	Physique nucléaire	R. SCHLICH	Géophysique marine
A. KOVACS	Physicochimie macromoléculaire	A. SKOLIOS	Physicochimie macromoléculaire
J. MARCHAL	Physicochimie macromoléculaire	M. VAN RESEMORTTEL	Viologie
E. MEYER	Mathématiques	A. VEILLARD	Chimie moléculaire
A.J.P. MEYER	Physique	A. ZUKER	Physique théorique

MAITRES DE RECHERCHE C.N.R.S. :

JOABBE	Physicochimie des Interfaces	G. MAIRE	Chimie
P. ALBRECHT	Chimie	A. MALAN	Physiologie respiratoire
E. ASLANIDES	Physique nucléaire et corpusculaire	R. MARC	Physicochimie et Macromoléculaire
F. BECK	Physique nucléaire et corpusculaire	R. MORAS	Physique nucléaire
J.P. BECK	Physiologie	Th. MULLER	Physique nucléaire et corpusculaire
M. BONHOMME	Physique nucléaire	G. MUNSCHY	Physique
H. BRAUN	Géologie	M. NAUCIEL-BLOCH	Physique des Solides
H. BRAUNSTEIN	Physique nucléaire	A. NICOLAIEFF	Viologie végétale
MCCADEVILLE	Chimie	H. PAQUET	Géologie
S. CANDAU	Physique des Solides	C. PLOT	Physicochimie macromoléculaire
M. CHAMPAGNE	Physique	L. PINCK	Biophysique
J.P. COFFIN	Biophysique	J. POIX	Chimie
A. CORET	Physique nucléaire et corpusculaire	J. POUYET	Chimie
M. CROISSIAUX	Physique	B. REES	Chimie
D. DUBOIS	Physique nucléaire et corpusculaire	P. REMY	Biochimie
J. DUBINGER	Physique nucléaire et corpusculaire	J. RINGEISEN	Physicochimie macromoléculaire
A. DURHAM	Géologie	J.P. RINGEISEN	Physicochimie macromoléculaire
F. DURST	Physiologie Végétale	P. SAUVAGE	Chimie
S. EL KMOSS	Physique	R. SCHANTZ	Physiologie Végétale
B. FRANCOIS	Physicochimie macromoléculaire	F. SCHERBLANG	Physique nucléaire et corpusculaire
M. FRANK-NEUMANN	Chimie organique	N. SCHULZ	Physique nucléaire et corpusculaire
E. FRANTA	Physicochimie macromoléculaire	C. SCHWAB	Physique
A. FRIDMANN	Physicochimie macromoléculaire	R. SELTZ	Physique nucléaire et corpusculaire
J. GALLOT	Physicochimie macromoléculaire	C. SIFFERT	Physique nucléaire et corpusculaire
J.P. GERBER	Physicochimie macromoléculaire	M. SIOCKEL	Géologie
J.B. GRAMAIN	Physicochimie macromoléculaire	Q. STRAZIELLE	Géologie
J. HERZ	Physicochimie macromoléculaire	M. SUFFERT	Physique nucléaire et corpusculaire
J. KIRCH	Physicochimie macromoléculaire	K. TRAORE	Physicochimie atomique et ionique
G. KOCH	Physicochimie macromoléculaire	R. VAROUI	Physicochimie macromoléculaire
E. KOCHANSKY	Physicochimie macromoléculaire	G. WABER	Physique nucléaire et corpusculaire
J. LAURENT	Physicochimie macromoléculaire	F. WEBER	Physique nucléaire et corpusculaire
P. LANG	Physicochimie macromoléculaire	J.P. WENIGER	Physique nucléaire et corpusculaire
C. LEVADIE	Physicochimie macromoléculaire	R. WITZ	Physique nucléaire et corpusculaire
F. LEVADIE	Physicochimie macromoléculaire	R. ZAN	Physique nucléaire et corpusculaire
A. LLORET	Physicochimie macromoléculaire	P. ZAN	Physique nucléaire et corpusculaire
B. LOTZ	Physicochimie macromoléculaire	L. ZILLIOX	Physique nucléaire et corpusculaire
B. LUU	Physicochimie macromoléculaire		

A mes parents,

A ma soeur,

A mes amis,

TABLE DES MATIERES

INTRODUCTION		
CHAPITRE I		
I. Le modèle de Drell-Yan		
A. Le modèle naïf		
B. Insuffisance du modèle		
II. Manifestations de l'annihilation $q\bar{q}$		
A. Influence de la cible		
B. Influence du faisceau		
C. Distribution angulaire		
III. Test de l'invariance d'échelle ("scaling")		
A. Le scaling dans les expériences de pions		
B. Le scaling dans les expériences de protons		
IV. Extraction de fonctions de structure		
V. Moment transverse du dilepton		
VI. Conclusions.		
CHAPITRE II : FILTRAGE EN LIGNE SUR L'EXPERIENCE NA10		
I. Présentation de l'expérience		
A. Résumé des motivations physiques		
B. Description de l'appareillage		
C. Simulation de l'expérience		
1) Paramètres du programme		
2) Acceptances		
D. Critères de filtrage		
II. Le filtrage en ligne : cas général		
A. Introduction		
B. Le filtrage en ligne		
1) Le volume des données		
2) Limitation de la sélectivité du déclenchement		
3) Le bruit de fond		
4) Conclusion		
C. Filtres logiciels (software)		41
1) Méthodes logicielles et matérielles (hardware)		
2) Amélioration des performances du logiciel		42
a) Programmation efficace		
b) Microprogrammation		
c) Adjonction d'opérateurs câblés		43
d) Multiprocessing		
e) "Self-made" processeur		
D. Les microprocesseurs par tranches		44
E. Filtres câblés		
III. Application à l'expérience NA10 : Description de la structure de filtrage adoptée.		45
1) Précision des calculs		
2) Vitesse		
3) Structure adoptée		
a) ordinateurs		
b) opérateurs câblés		
c) durée de l'opération		46
d) description de l'ensemble de filtrage		
IV. La micromachine de GESPRO		
A. La partie opérative de la micromachine		48
1) Description de l'unité CPE		
a) l'opérateur arithmétique et logique		
b) les registres		
c) les indicateurs		
d) les multiplexeurs		
e) la logique de décodage		
2) La commande de l'unité de traitement		51
B. La partie contrôle de la micromachine		
1) L'unité de contrôle		52
2) La gestion du séquençement		
a) les sauts inconditionnels		53
b) les sauts conditionnés par la valeur d'un indicateur		53
c) les sauts d'éclatement selon les contenus du bus PX ou du tampon PR		
3) Le contrôle des indicateurs (FC_0-FC_3).		54

CHAPITRE III : Logiciel	55	
I. Les instructions		
A. Généralités		
B. Format général d'une instruction	57	
C. Déroulement d'une instruction	59	
a) cas particulier de l'architecture utilisée		c) dépendance du temps d'accès d'une cellule avec l'adresse 85
b) définition du jeu d'instructions		d) ambiguïté d'adresse 87
D. Outils de génération d'instructions	62	
E. Le jeu d'instructions	64	
a) les instructions à référence mémoire		
b) les instructions à référence interne	66	
c) les débranchements et contrôles	66	
d) les instructions de traitement des indicateurs		
e) le traitement des interruptions	67	
f) les instructions "système"		
1. L'initialisation de Gespro		
2. Le changement de contexte		
II. Le système de développement de programmes	68	
A. Introduction		
B. Généralités		
C. Structure du système	72	
a) gestion du fichier de définition d'instructions		
1. Introduction		
2. Caractéristiques générales		
b) Assembleur	74	
III. Le système d'aide à la microprogrammation		
A. Introduction		
B. Choix du langage de microprogrammation	75	
C. Présentation du système		
CHAPITRE IV : Test de l'appareillage	77	
I. Introduction		
II. Test des opérateurs câblés par NORD-10		
III. Test des opérateurs câblés par GESPRO	79	
IV. Les tests mémoire		
A. Généralités sur les mémoires		
B. Défauts testés	83	
a) absence de cellule (s)		
b) perturbation d'une cellule par l'état de ses voisines	85	
C. Test NORD 10 et GESPRO		
D. Performances des microprogrammes de test		
CONCLUSION		89
ANNEXE		91
REFERENCES		97
REMERCIEMENTS		99

I N T R O D U C T I O N

Notre travail s'inscrit dans le cadre de l'expérience NA10, dont l'objet est l'étude de la production inclusive de dimuons de haute masse dans les collisions hadroniques, en particulier les collisions pion-nucléon. Cette expérience permet de tester :

- le modèle de Drell et Yan, qui décrit le processus de production
- certains aspects de la structure des hadrons

Dans le chapitre 1 nous décrivons le modèle de Drell-Yan de base, ses prédictions que nous confronterons avec les résultats expérimentaux les plus récents.

Dans le chapitre 2 nous aborderons le problème du filtrage en ligne des événements dans une expérience de physique des hautes énergies : d'abord d'un point de vue général, puis nous nous attacherons plus spécialement au cas de l'expérience NA10 réalisée auprès du SPS du CERN à Genève. Nous décrivons l'expérience, puis présenterons sa simulation par des méthodes du type Monte-Carlo, puis les résultats de cette simulation et leurs implications quant aux critères de filtrage des événements en ligne et à la réalisation du filtre. C'est un filtre multiprocesseur (comportant 4 exemplaires de "GESPR0" : micro-processeur par tranches auquel est adjoit un module "multiplicateur" qui peut calculer rapidement la masse des candidats dimuons et décider de la validité de l'événement) s'appuyant sur les données de modules de recherche du plus grand moment transverse qui constituent un pré-filtrage. GESPR0 est micro-programmable et programmable : nous traiterons de l'aspect logiciel du filtrage dans le chapitre 3 en particulier :

- la génération d'instructions
- le système de développement de programmes
- le système d'aide à la microprogrammation

Dans le chapitre 4 nous décrivons les tests de l'appareillage, en particulier les microprogrammes et programmes de test des opérateurs câblés ; nous insisterons sur la mise en oeuvre de séquences de tests mémoire.

C H A P I T R E I

PRODUCTION DE DILEPTONS DANS LES COLLISIONS HADRONIQUES

I. LE MODELE DE DRELL-YAN^{1,3,4,5}

A. Le modèle naïf

Dans ce modèle la production de paires de leptons de haute masse, dans les collisions de deux hadrons A et B :

$$A + B \rightarrow \ell^+ + \ell^- + X$$

est décrite comme un processus électromagnétique d'annihilation d'un quark de l'un des hadrons avec un antiquark de l'autre hadron, produisant un photon du genre temps. Ce photon se désintègre en une paire de leptons, processus illustré par la figure I.1, où sont également indiquées les variables cinématiques.

Les quarks restants se manifestent sous forme de hadrons dans l'état final.

La section efficace de production de dileptons est obtenue en sommant de façon incohérente tous les processus élémentaires du type :

$$q_i + \bar{q}_i \rightarrow \ell^+ + \ell^-$$

pondérés par la probabilité de trouver dans les hadrons considérés le quark et l'antiquark qui vont s'annihiler.

Cette section efficace s'écrit :

$$\sigma(A + B \rightarrow \ell^+ + \ell^- + X) = \int \int dx_1 dx_2 \sum_i [f_i^A(x_1) \cdot f_i^B(x_2) + \bar{f}_i^A(x_1) \cdot \bar{f}_i^B(x_2)] \cdot \sigma(q_i + \bar{q}_i \rightarrow \ell^+ + \ell^-) \cdot \delta[q^2 - (k_i + \bar{k}_i)^2] dq^2$$

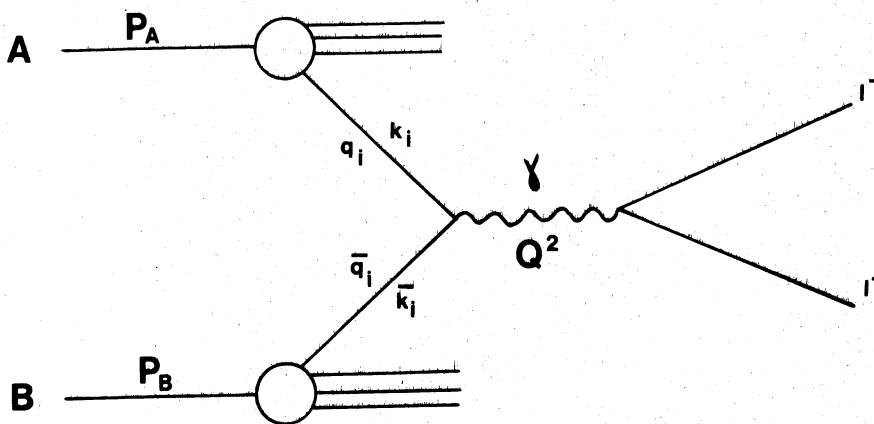


Fig.I.1 : Processus de Drell-Yan

où x_1 et x_2 se réfèrent aux hadrons A et B respectivement.

Dans le modèle simple, les quarks constituant les hadrons ont de faibles moments transverse k_T et les fonctions probabilités f et \bar{f} dépendent uniquement des quantités sans dimension x_1 et x_2 qui sont les fractions du moment initial portées par chacun des quarks constituants :

$$k_i = x_1^i P_A \quad \bar{k}_i = x_2^i P_B$$

Soit P l'impulsion des hadrons A et B dans le centre de masse ; dans l'hypothèse des grandes impulsions ($P \rightarrow \infty$) on néglige les masses des hadrons et des quarks initiaux. L'énergie disponible dans le centre de masse s'écrit :

$$S = (P_A + P_B)^2 = (2P)^2 = 4P^2$$

L'énergie de la paire s'écrit :

$$Q_0 = (x_1 + x_2) P$$

L'impulsion de la paire le long du faisceau s'écrit

$$P_L = (x_1 - x_2) P$$

L'impulsion transverse de la paire est égale à la somme des impulsions transverse de quarks, elle est négligeable dans l'hypothèse $P \rightarrow \infty$.

La masse invariante de la paire s'écrit :

$$m^2 = Q_0^2 - P_L^2 = 4 x_1 x_2 P^2 = x_1 x_2 S$$

Les deux paramètres directement mesurables sont :

$$i) \text{ la quantité } \tau \text{ définie par : } \tau = \frac{m^2}{S}$$

Dans la limite des hautes énergies, τ s'écrit :

$$\tau = x_1 x_2$$

ii) la quantité X_F dite variable de Feynman définie par

$$X_F = \frac{2P_L}{\sqrt{S}} = \frac{2(x_1 - x_2)P}{2P} = x_1 - x_2$$

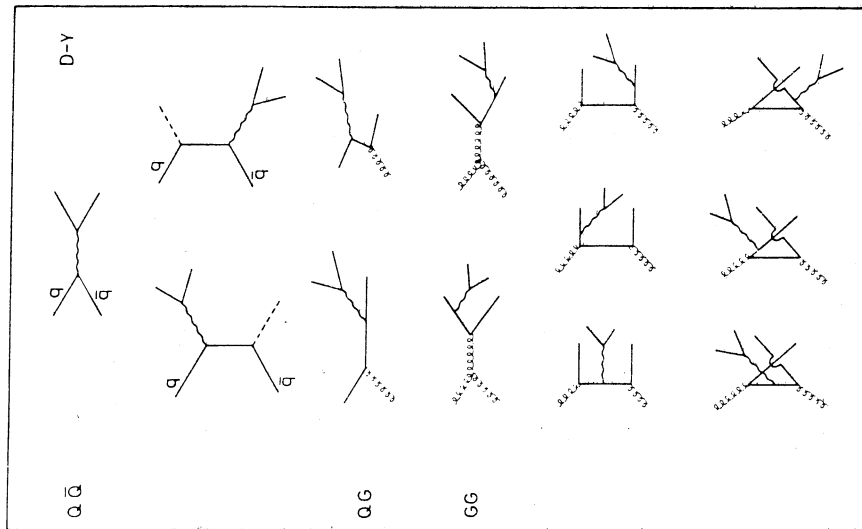


Fig.I.2 : Diagrammes de production d'un dilepton

Le couplage $\gamma q_i \bar{q}_i$ étant supposé ponctuel, les quarks ayant un spin 1/2 et une charge e_i , la section efficace peut s'écrire sous la forme suivante :

$$m^3 \frac{d^2\sigma}{dm dx_F} = \frac{8\pi\alpha^2}{3N_C} \cdot \frac{1}{\sqrt{x_F^2 + 4\tau}} \cdot \sum e_i^2 [g_i^A(x_1) \bar{g}_i^B(x_2) + \bar{g}_i^A(x_1) g_i^B(x_2)]$$

Les fonctions $g(x)$ sont les fonctions de structure définies par

$$g(x) = x \cdot f(x)$$

où $f(x)$ sont les probabilités définies plus haut

Les prédictions du modèle sont les suivantes :

a) La section efficace doit être invariante d'échelle

$$m^3 \frac{d^2\sigma}{dm dx_F} \text{ et } m^3 \frac{d\sigma}{dm} \text{ sont des fonctions universelles de la variable sans}$$

dimension τ et ne dépendent pas de l'énergie

b) La section efficace est inversement proportionnelle au nombre de couleurs N_C (si les quarks sont de N_C couleurs différentes leur taux d'annihilation est réduit d'un facteur N_C par rapport à des quarks incolores). Les fonctions $g(x)$ ne dépendent pas de la couleur.

c) La distribution angulaire d'un lepton dans le système du dilepton doit être de la forme ⁶⁾ :

$$\frac{d\sigma}{d\Omega} = K (1 + \cos^2\theta)$$

d) La section efficace (en forme et en amplitude) est déterminée à partir des fonctions de distribution des quarks extraites des expériences de diffusion profondément inélastique de leptons, par exemple :

$$e + N \rightarrow e + X$$

$$\nu_\mu + N \rightarrow \mu^- + X$$

e) L'impulsion transverse k_T des quarks dans les hadrons étant supposée faible, le moment transverse du dilepton doit être faible également.

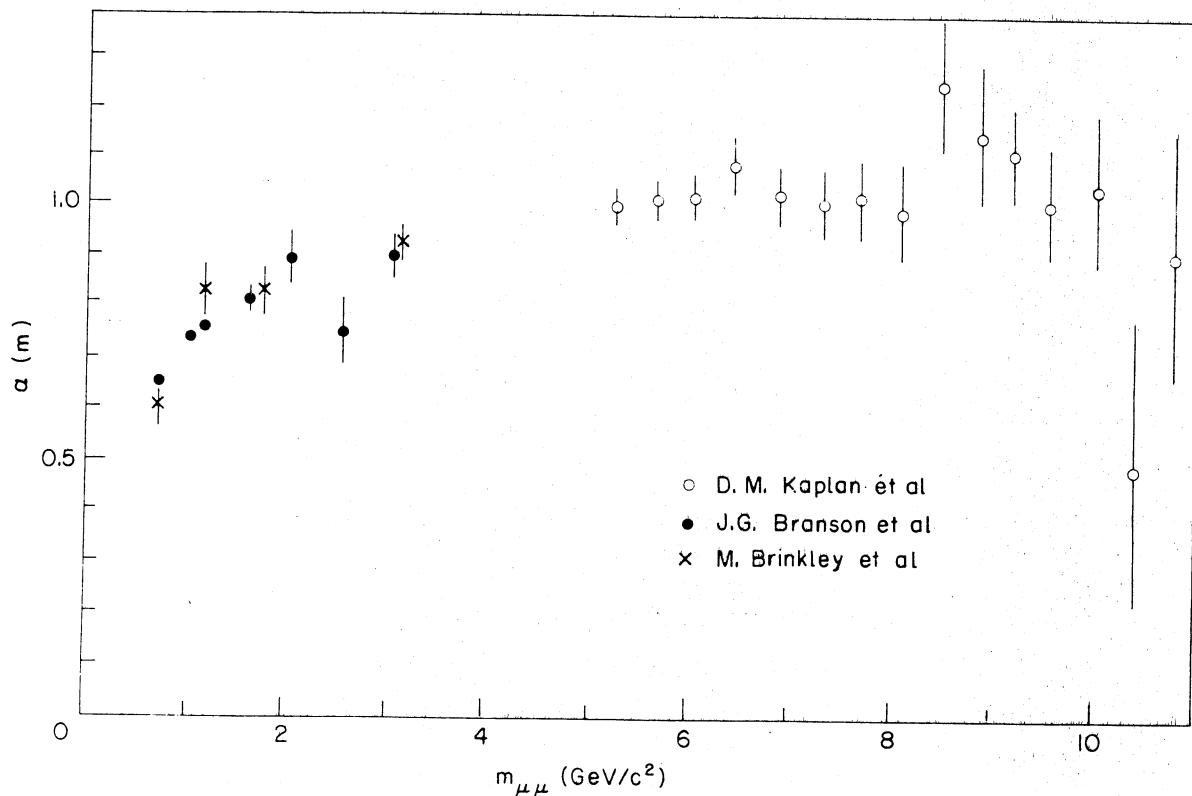


Fig.I.3 : Dépendance de la section efficace en fonction du nombre de nucléons A de la cible

f) Finalement, on s'attend à deux jets hadroniques (produits par les fragments des hadrons initiaux) accompagnant la paire de leptons.

B. Insuffisance du modèle

Il y a deux raisons principales d'aller plus loin que le modèle naïf présenté plus haut :

* la violation de l'invariance d'échelle observée dans les expériences de diffusion profondément inélastique, nécessitant l'introduction d'une dépendance en q^2 des fonctions de structure.

* Les grands moments transverses des dileptons apparaissant expérimentalement qui sont en contradiction flagrante avec le modèle naïf de Drell et Yan.

Il y a également des raisons d'ordre purement théorique qui invitent à aller plus loin que le simple graphe d'annihilation ; on ne peut négliger entièrement la contribution de graphes d'ordre plus élevé, à la production de dileptons, du type quark-antiquark, quark-gluon ou gluon-gluon (illustrés sur la figure I.2).

II. MANIFESTATIONS DE L'ANNIHILATION $q\bar{q}$ *

A. Influence de la cible

Dans le processus de Drell-Yan, il est postulé que les centres diffuseurs dans les nucléons agissent de façon incohérente. Il est donc raisonnable d'attendre un comportement linéaire de la section efficace totale en fonction du nombre atomique A de la cible. Pour mémoire rappelons que le comportement, dans les processus hadroniques, de la section efficace est de la forme :

$$\sigma = A^{2/3}$$

La figure I.3 montre la dépendance du nombre α en fonction de la masse de dimuons produits à l'aide d'un faisceau de protons. α caractérise une section efficace supposée de la forme

$$\sigma = A^\alpha$$

* références 2, 7, 8, 9, 10) pour les résultats expérimentaux

Jusqu'à 4 GeV/c² le comportement correspond à une dépendance $\alpha = 2/3$; au-dessus à un comportement compatible avec $\alpha = 1$, ce qui indique un changement de régime dans la production de dimuons.

Dans les réactions induites par des pions, l'expérience Chicago-Princeton a trouvé un exposant $\alpha = 1.12 \pm 0.05$ au dessus de 4 GeV/c², mais les résultats récents de l'expérience NA3 supportent l'idée d'une interaction incohérente des quarks, avec une valeur : $\alpha = 1.03 \pm 0.03$. Cette différence cause quelques difficultés si l'on veut comparer les deux expériences.

B. Influence du faisceau

L'annihilation concerne un quark d'un des hadrons et un antiquark de l'autre hadron. La cible est constituée de :

- neutrons dont la constitution en quarks de valence est : udd
 - protons dont la constitution en quarks de valence est : uud
- pour ces particules les antiquarks sont dans la mer exclusivement.

Par contre la distribution en quarks et antiquarks des particules projectiles peut être plus variée, en utilisant les $\pi^-(ud)$, $\pi^+(u\bar{d})$, $\bar{p}(\bar{u}\bar{u}\bar{d})$, $K^-(\bar{u}s)$, etc..

Les relations suivantes peuvent être démontrées (elles peuvent être comprises intuitivement en examinant la structure en quarks des différentes particules concernées) :

$$\begin{aligned} \sigma(\pi^+p) &= \sigma(K^+p) > \sigma(\pi^+n) > \sigma(K^+n) > \sigma(\pi^-p) > \sigma(K^-p) \\ \sigma(\pi^-p) &= \sigma(K^-p) > \sigma(\pi^+n) > \sigma(K^+n) > \sigma(K^0p) = \sigma(K^0n) \end{aligned}$$

L'expérience NA3 a obtenu des données avec des faisceaux de π^+ , π^- , K^+ , K^- , p et \bar{p} . La figure I.4 donne le spectre de masse obtenu avec des faisceaux de π^+ , π^- , K^+ , K^- de 200 GeV/c normalisés au même flux de particules incidentes. Les résultats sont compatibles avec :

$$\begin{aligned} \sigma(\pi^+p) &\approx \sigma(K^+p) \\ \sigma(\pi^-p) &\approx \sigma(K^-p) \end{aligned}$$

La relation entre $\sigma(\pi^+p)$ et $\sigma(\pi^+n)$ est la suivante pour une cible isoscalaire (comportant autant de quarks u que de quarks d, c'est à dire autant de protons que de neutrons, par exemple : carbone, deutérium, etc...) :

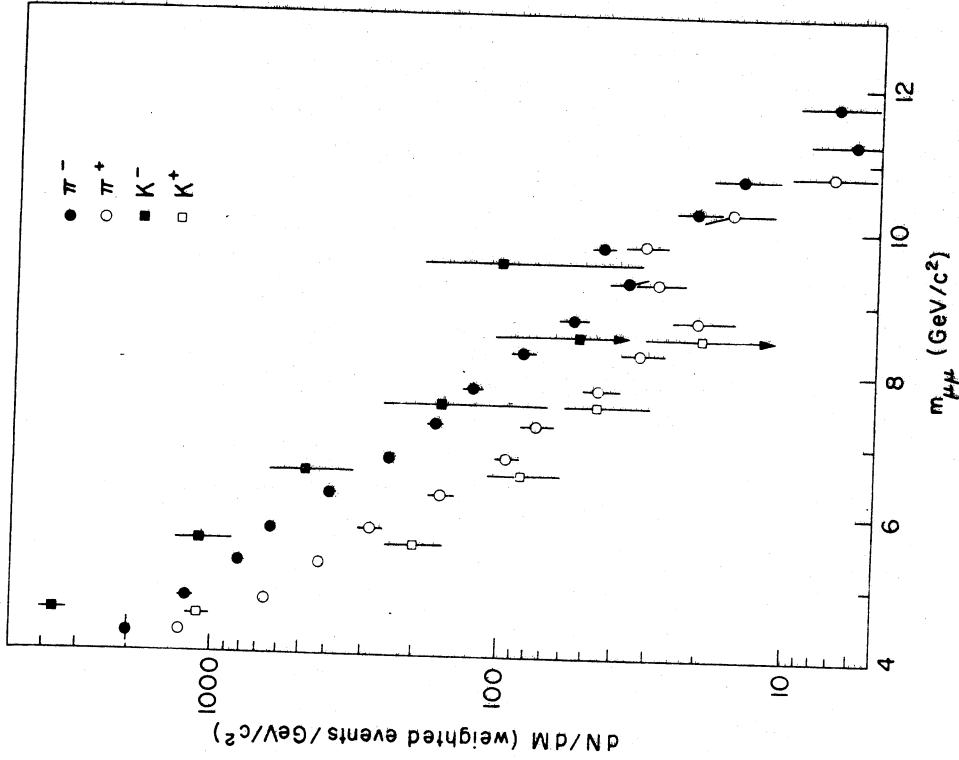


Fig.I.4 : Sections efficaces de production avec des faisceaux de π^+ , π^- , K^+ , K^-

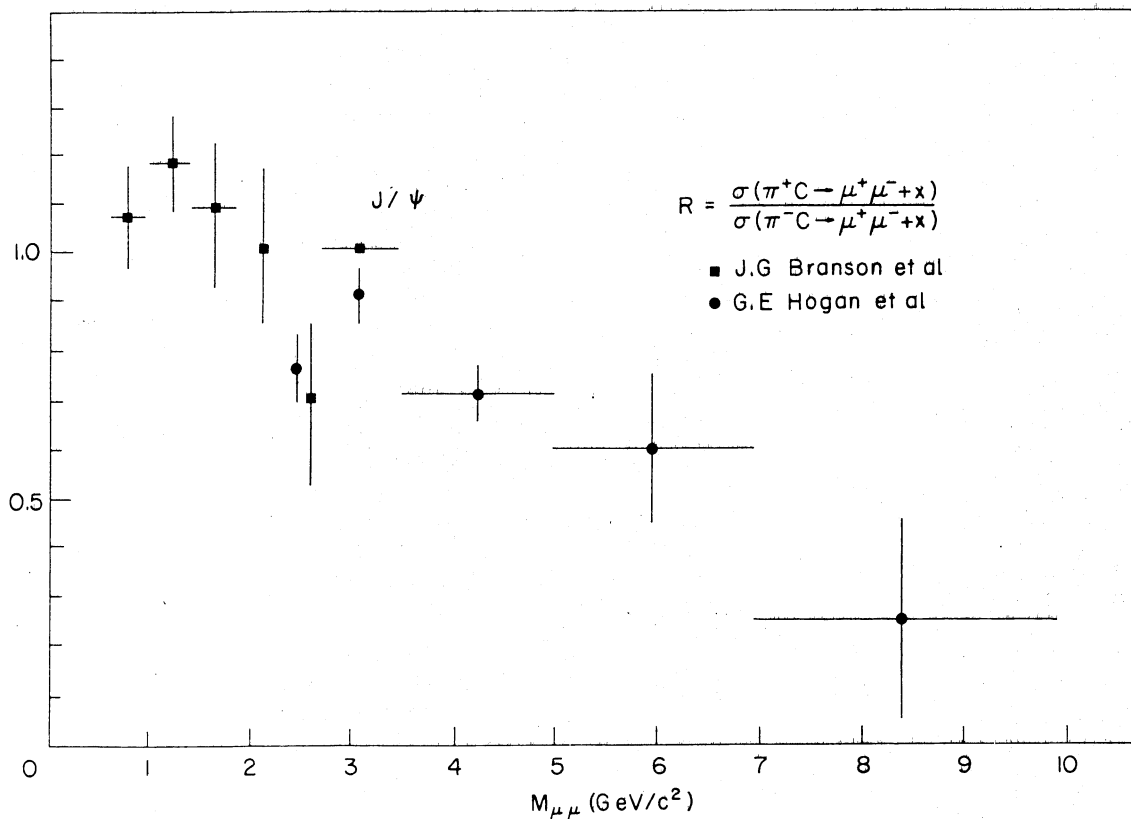


Fig.I.5 : Rapport de production π^+/π^-

$$\frac{\sigma(\pi^+N)}{\sigma(\pi^-N)} = \frac{e_d^2 \frac{d(x) \bar{d}(x)}{2} \frac{1}{u(x) \bar{u}(x)}}{e_u^2} = \frac{1}{4}$$

Ceci est vrai dans la limite où on ne tient pas compte des quarks de la mer. La figure I.5 illustre les résultats expérimentaux obtenus avec une cible de carbone. Il y a clairement une transition entre un rapport 1 attendu pour la production hadronique et un rapport compatible avec 0.25 pour des masses élevées de dimuons.

Le résultat obtenu pour le rapport :

$$\frac{\sigma(\pi^+N)}{\sigma(pN)}$$

est très remarquable comme on peut le voir sur la figure I.6 : ce rapport égal à 1 pour de faibles masses atteint 200 pour des masses de $10 \text{ GeV}/c^2$. Ceci s'explique aisément par l'existence d'anti-quarks de valence dans le π^- , alors que pour le proton les anti-quarks n'existent que dans la mer, donc ne portent qu'une faible fraction du moment. Le test idéal de cette idée est la comparaison entre $\sigma(pN)$ et $\sigma(\pi^-N)$. Ce rapport est aussi montré sur la figure I.6.

C. Distribution angulaire ⁶⁾

Dans le cadre du modèle simple d'annihilation, la distribution angulaire d'un lepton dans le système du dilepton est typique de la désintégration d'un état $J^\pi = 1^-$, c'est à dire de la forme :

$$\frac{d\sigma}{d\Omega} = A (1 + \cos^2\theta)$$

θ est mesuré à partir de la direction définie par $q\bar{q}$; si les quarks n'ont pas de moment transverse les hadrons initiaux définissent cette direction.

Les résultats des expériences Chicago-Princeton et NA3 utilisant des pions et de l'expérience CHMNP utilisant des protons confirment une dépendance en $1 + \cos^2\theta$ en dehors des résonances alors qu'elle est isotrope dans la région du J/ψ et du Υ . Au delà du simple modèle de Drell-Yan, il est prévu qu'à grand x_F , le photon intermédiaire peut acquérir une polarisation longitudinale, qui induit une distribution angulaire du type $\sin^2\theta$. Les résultats récents de l'expérience Chicago-Princeton tendent à confirmer cette idée.

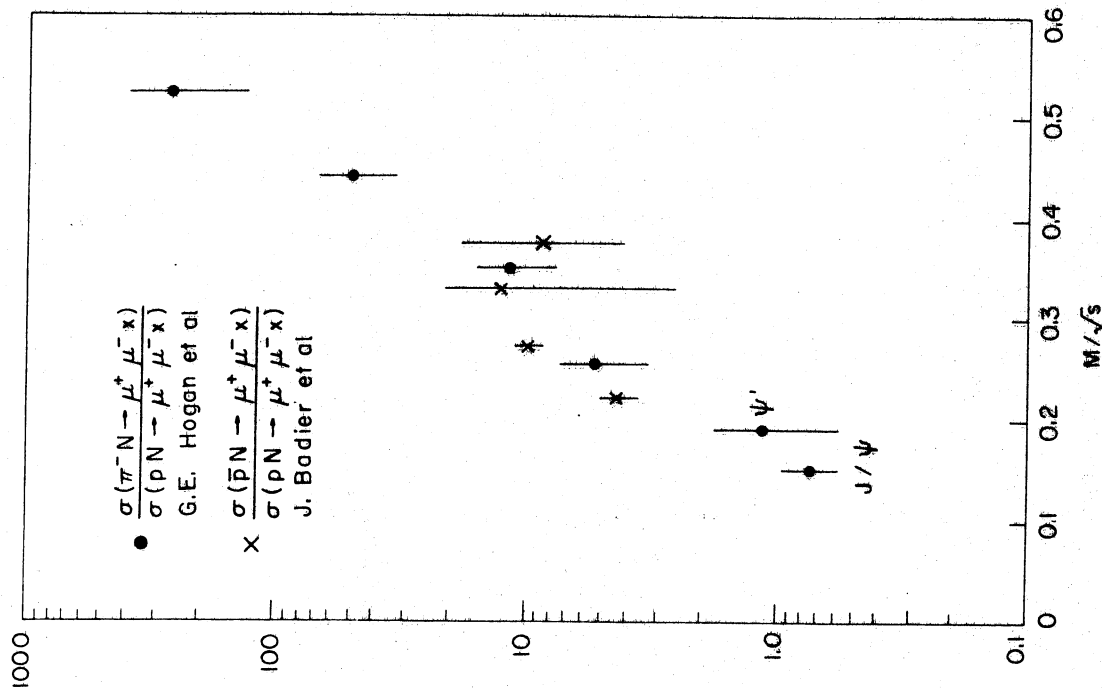


Fig. I.6 : Rapports de production π^-/p et \bar{p}/p

III. TEST DE L'INVARIANCE D'ECHELLE ("SCALING")

La loi d'invariance d'échelle

$$m^3 \frac{d\sigma}{dm} = F(\tau)$$

est une des prédictions principales du modèle de Drell-Yan. Une de ses conséquences est la suivante : les fonctions de distribution des quarks dans les hadrons ne dépendent que de la variable sans dimension x . Les violations de l'invariance d'échelle peuvent être générées dans la théorie en introduisant dans les fonctions de structure des facteurs de forme dépendant de q^2 .

Des violations de scaling ont été observées dans les expériences de diffusion profondément inélastique en muons et neutrinos (de l'ordre de 20 %), mais ces violations ne sont importantes que pour de faibles valeurs de q^2 ($q^2 < 20 \text{ GeV}^2/c^2$) et à grand x ($x > 0.4$). Il n'y a jusqu'à maintenant aucune évidence expérimentale de violation sensible de scaling dans des processus à $q^2 > 20 \text{ GeV}^2/c^2$.

A. Le scaling dans les expériences de pions

Sur la figure I.7 nous avons représenté la quantité $m^3 d\sigma/dm$ obtenue par différentes expériences (NA3, Chicago-Princeton et WA11) en fonction de $\sqrt{\tau} = m / \sqrt{s}$ pour des faisceaux de pions de 150, 225 et 280 GeV/c. L'accord entre les points à 150 et 280 GeV/c est relativement bon. La précision obtenue dans chacune des expériences n'est jamais meilleure que 20 %, ce qui veut dire qu'il ne faut pas être surpris outre mesure si la comparaison entre différentes expériences, comportant des erreurs systématiques différentes, ne permet pas de tester la loi d'invariance d'échelle à mieux que 50 % de précision. La différence entre l'expérience Chicago-Princeton, dont les résultats correspondent à une énergie intermédiaire (pions de 225 GeV/c), et les deux autres, semble être le reflet de la dépendance en A de la section efficace. Il est intéressant de noter que si l'on utilise une dépendance en A^1 pour les trois expériences, on retrouve l'invariance d'échelle, alors que si l'on utilise une dépendance en $A^{1.12}$, pour les trois expériences, on introduit une apparente violation de l'invariance d'échelle.

Il faut remarquer que ces tests ne sont de toute façon pas vraiment probants à cause du domaine cinématique restreint exploré. En effet, on a :

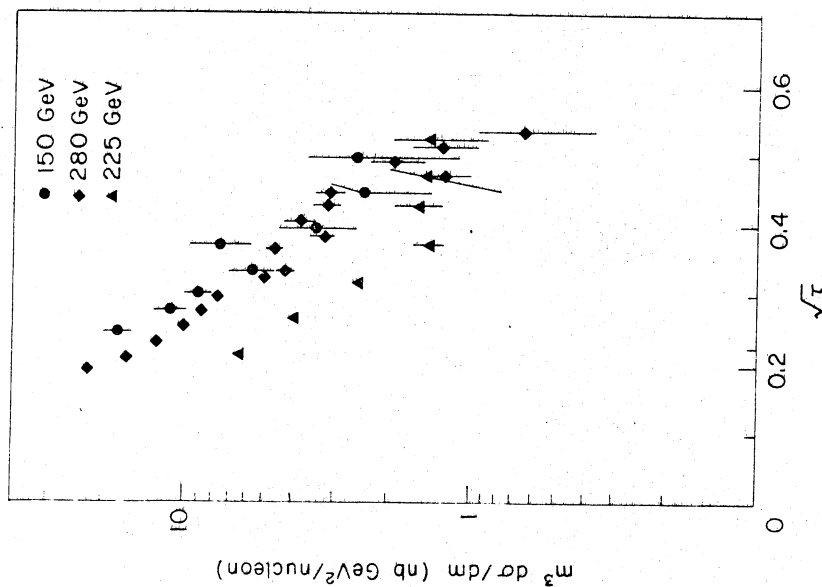


Fig. I.7 : Invariance d'échelle dans les expériences induites par des π^-

$$17.3 \leq \sqrt{s} \leq 23.7 \text{ GeV}/c$$

ce qui signifie que la région de recouvrement en q^2 s'étend à :

$$15 < q^2 < 25 \text{ GeV}^2 \text{ à } \sqrt{\tau} = 0.05$$

$$70 < q^2 < 140 \text{ GeV}^2 \text{ à } \sqrt{\tau} = 0.25$$

Si l'on se base sur les résultats obtenus en diffusion profondément inélastique, on ne doit pas s'attendre à des violations spectaculaires de scaling dans ces domaines de q^2 .

B. Le scaling dans les expériences de protons.

Un test similaire de l'invariance d'échelle a été effectué par la collaboration CFS (Columbia-FNAL-Stony Brook) avec des faisceaux de protons de 200, 300 et 400 GeV/c. Dans cette analyse l'invariance d'échelle est vérifiée à 20 % près, mais on peut faire le même commentaire que précédemment : l'invariance d'échelle est testée dans une zone de q^2 suivante :

$$16 \leq q^2 \leq 32 \text{ GeV}^2$$

pour la valeur la plus faible de $\sqrt{\tau} = 0.04$ et une zone

$$80 \leq q^2 \leq 160 \text{ GeV}^2$$

pour la valeur la plus élevée de $\sqrt{\tau} = 0.2$

On dispose aussi de données obtenues aux ISR correspondant à $\sqrt{s} = 62 \text{ GeV}$, ce qui élargit le domaine en q^2 disponible. Mais en pratique le test est limité aux masses entre les résonances J/ψ et τ , ce qui signifie que les résultats de FNAL et des ISR se recouvrent dans un domaine en τ très étroit.

La figure I.8 représente la quantité :

$$n^3 \frac{d^2\sigma}{dm dx} \bigg|_{x=0} = F(\tau)$$

mesurée dans les expériences CFS et CHMP. La contribution principale de l'expérience réalisée aux ISR est la donnée de la fonction $F(\tau)$ pour de faibles valeurs de τ , mais le test de l'invariance d'échelle est limité à la zone $\tau = 0.04$ qui correspond à la limite inférieure des données FNAL et

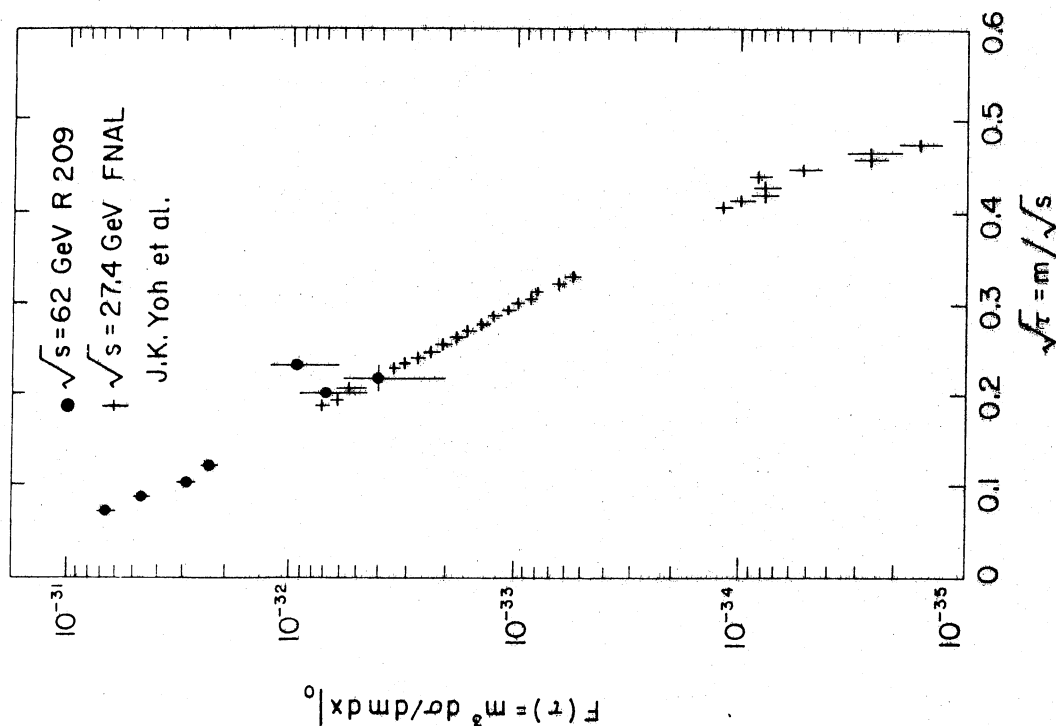


Fig.1.8 : Invariance d'échelle dans les expériences induites par des p

à la limite supérieure des données des ISR (quelques événements au-dessus du τ). Ce recouvrement montre que l'invariance d'échelle est vérifiée à 50 % près dans une zone telle que

$$16 < q^2 < 150 \text{ GeV}^2/c^2$$

Mais ce test est effectué à $x_F = 0$ donc $x_1 = x_2 = \sqrt{\tau} = 0.2$ ce qui correspond à une région où la théorie QCD ne prévoit pratiquement pas de violation de scaling. Mais la leçon de la comparaison est ailleurs : $F(\tau)$ ainsi mesurée entre $\sqrt{\tau} = 0.05$ et 0.5 peut être extrapolée de la région explorée aux ISR à celle étudiée à FNAL.

IV. EXTRACTION DE FONCTIONS DE STRUCTURE

Nous avons vu que dans le modèle "naïf" de Drell et Yan la section efficace de production de dimuons est liée aux fonctions de structure en quarks des hadrons. Les mesures faites dans les expériences de diffusion profondément inélastique s'expriment aussi en termes de fonctions de distribution de quarks. Si l'idée de constituants dans les hadrons est pertinente, on peut espérer que la distribution de ces constituants est indépendante de la manière de la mesurer. Si le modèle des partons est pertinent, l'idée d'universalité implique que les fonctions de structure déterminées via le modèle de Drell-Yan et celles déterminées via la diffusion profondément inélastique sont identiques.

Fonction de structure des pions

Ceci est une contribution unique des expériences dileptons car il est très difficile de tester les constituants des pions autrement. Jusqu'à présent l'analyse la plus complète est celle de l'expérience NA3. Il est avantageux d'utiliser des faisceaux de pions car ces derniers comportent des antiquarks de valence ; de plus on peut négliger la contribution des quarks de la mer. Ainsi la section efficace s'écrit :

$$\frac{d^2\sigma}{dx_1 dx_2} \propto \frac{1}{x_1^2 x_2^2} V(x_1) G(x_2)$$

où x_1 et x_2 sont extraits des quantités mesurables

$$m^2 = x_1 x_2 s$$

$$x_F = x_1 - x_2$$

Il n'y a qu'une fonction de distribution pour les quarks de valence à l'intérieur du pion :

$$V(x_1) = \bar{u}_V^\pi(x_1) = d_V^\pi(x_1) = u_V^+(x_1) = d_V^+(x_1)$$

Le premier test consiste à vérifier que la section efficace en fonction de x_1 et x_2 se présente bien sous la forme :

$$\text{fonction}(x_1) \times \text{fonction}(x_2)$$

Mais le test n'est pas entièrement satisfaisant, ce qui peut provenir du fait qu'on néglige la contribution des quarks de la mer du pion. Il existe une façon d'éliminer la contribution de la mer en soustrayant les sections efficaces des π^- et des π^+ après normalisation. En même temps on élimine complètement la contribution de la mer des nucléons. Dans le cas d'une cible de platine on trouve :

$$\left[\frac{d^2\sigma}{dx_1 dx_2} \right]_{\pi^- p \text{ Pt}} - \left[\frac{d^2\sigma}{dx_1 dx_2} \right]_{\pi^+ p \text{ Pt}} \propto \frac{1}{x_1^2 x_2^2} V(x_1) \frac{1}{9} [u_V(x_2) + 2d_V(x_2)]$$

$$\text{où : } \begin{aligned} u_V(x) &= u^p(x) = d^n(x) \\ d_V(x) &= d^p(x) = u^n(x) \end{aligned}$$

La factorisation est beaucoup mieux vérifiée et on peut extraire la forme de la fonction de structure du pion, illustrée par la figure I.9.

$$V(x) = x^{0.4 \pm 0.06} (1-x)^{0.9 \pm 0.06}$$

Cette distribution est plus plate que dans le cas des nucléons. Cette analyse donne également la composante de la mer du pion et, en étudiant la fonction $G(x_2)$, les fonctions de structure du nucléon.

Pour le nucléon le résultat est le suivant :

$$\begin{aligned} u_V(x) &= 10.5 x^{1.02} (1-x)^{4.04} \\ d_V(x) &= 6.3 x^{1.02} (1-x)^{5.04} \\ S(x) &= 0.35 (1-x)^{6.0} \end{aligned}$$

La forme des fonctions de distribution des quarks de valence s'accorde bien avec celle obtenue en diffusion profondément inélastique de neutrinos,

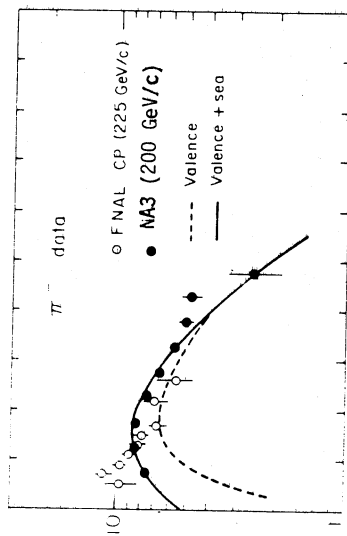


Fig.I.9 : Fonction de structure du pion

mais leur intégrale est deux fois plus grande, ce qui peut être dû à une extrapolation sensible aux faibles valeurs de x inobservées.

V. MOMENT TRANSVERSE DU DILEPTON

C'est le point sur lequel l'expérience entre le plus en conflit avec le modèle naïf. Dans le modèle de base le moment transverse du dilepton est la somme des moments transverses des deux quarks qui s'annihilent. A cause du confinement on s'attend à un moment transverse intrinsèque k_T du quark de l'ordre de 300 MeV/c indépendant de l'énergie du faisceau ; ce qui donne un P_T du dilepton de l'ordre 500 MeV/c. Les données obtenues jusqu'à présent en πN diffèrent des prévisions :

- i) le P_T moyen du dilepton (plus précisément du dimuon) augmente avec sa masse jusqu'à une masse de 4 GeV/c² pour atteindre ensuite un plateau
- ii) La valeur du plateau augmente avec s . Ceci est illustré sur la figure I.10

L'augmentation du P_T a été paramétrisée de la façon suivante :

$$\langle P_T \rangle = (600 + 22 \sqrt{s}) \text{ MeV/c}$$

Qualitativement on peut voir l'origine des phénomènes à grand $\langle P_T \rangle$ dans les diagrammes d'ordre plus élevé où le dilepton de grand $\langle P_T \rangle$ est compensé, par exemple, par un jet de gluon.

$\langle P_T \rangle$ ne semble pas dépendre de x_F .

VI. CONCLUSIONS

Nous avons vu les succès qualitatifs du modèle de Drell-Yan, qui affermissent l'idée de constituants à l'intérieur des hadrons, avec les nombres quantiques des quarks. Nous avons vu également qu'il est nécessaire d'aller au delà du modèle naïf ; l'étude de la production de dileptons est un domaine intéressant pour tester les corrections QCD. Mais ces corrections affectent seulement des régions cinématiques étroites, ou donnent une renormalisation générale, expliquant pourquoi les prédictions du modèle simple sont bien satisfaites pour la plus grande partie des données.

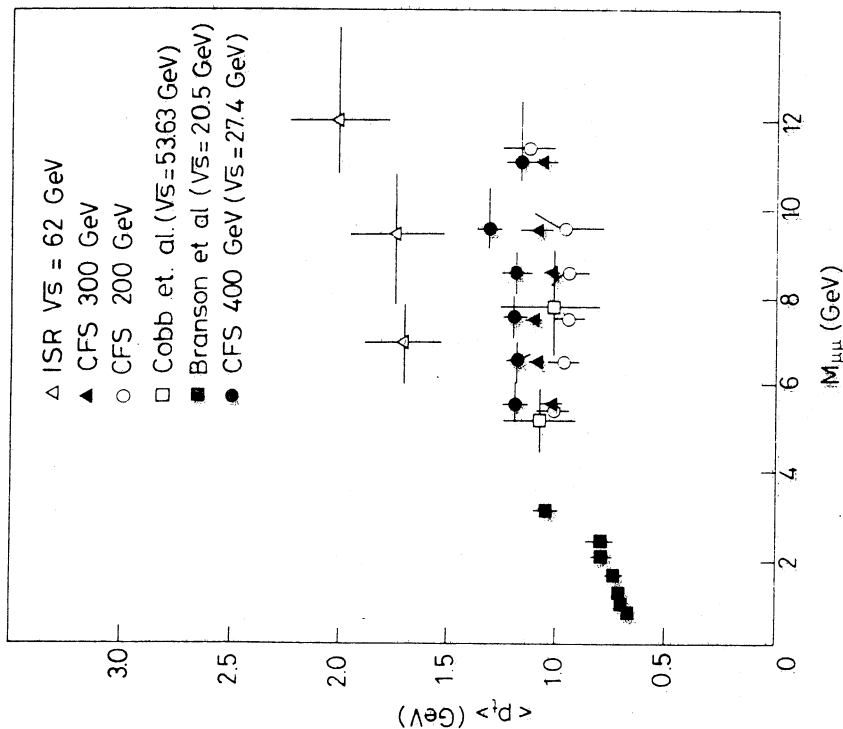


Fig.I.10 : distribution du $\langle P_T \rangle$ à différentes énergies

Beaucoup de résultats expérimentaux nouveaux ont été obtenus ces dernières années donnant une interprétation satisfaisante du processus.

Des progrès peuvent être faits grâce à une statistique plus grande, mais les sections efficaces chutent dramatiquement ; un ordre de grandeur de gain dans la statistique permet seulement d'étendre la plage de masse de 2 ou 3 GeV/c^2 . Ainsi il ne faut pas s'étonner si le progrès est moins rapide maintenant.

CHAPITRE II

FILTRAGE EN LIGNE SUR L'EXPERIENCE NA10

I. PRESENTATION DE L'EXPERIENCE

A. RESUME DES MOTIVATIONS PHYSIQUES

L'expérience NA10 permet d'étudier la production inclusive de paires de muons massives (au-dessus de 4 GeV/c²) par la réaction :

$$\pi^+ p \rightarrow \mu^- \mu^+ + X$$

ceci à l'aide d'un faisceau intense de pions, jusqu'à 10¹⁰ π par pulse, avec un appareillage permettant :

- une grande acceptance (voir fig.II.1)
- une haute résolution en masse (voir fig.II.2)

Les buts de l'expérience sont les suivants :

- (i) l'étude du continuum $\mu^- \mu^+$ jusqu'à des masses de 18 GeV/c² environ, en particulier la dépendance en impulsion transverse P_t et en variable de Feynman X_F , la vérification de l'invariance d'échelle dans une zone d'impulsion des pions incidents assez large (impulsion dans le système du laboratoire entre 100 et 300 GeV/c)

- (ii) l'investigation des fonctions de structure pioniques : Si les données expérimentales confirment la validité du modèle de Drell-Yan, ces fonctions peuvent être déduites des distributions en M^2/s et en X_F des dimuons

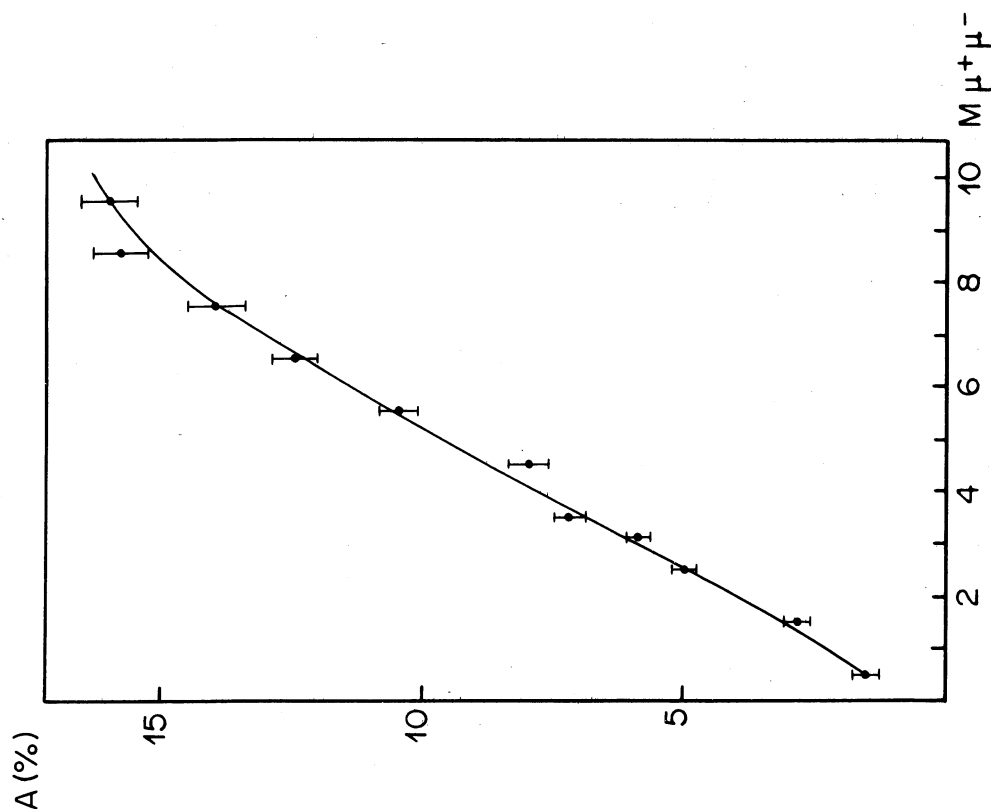


Fig.II.1 : Acceptance en masse de l'appareillage

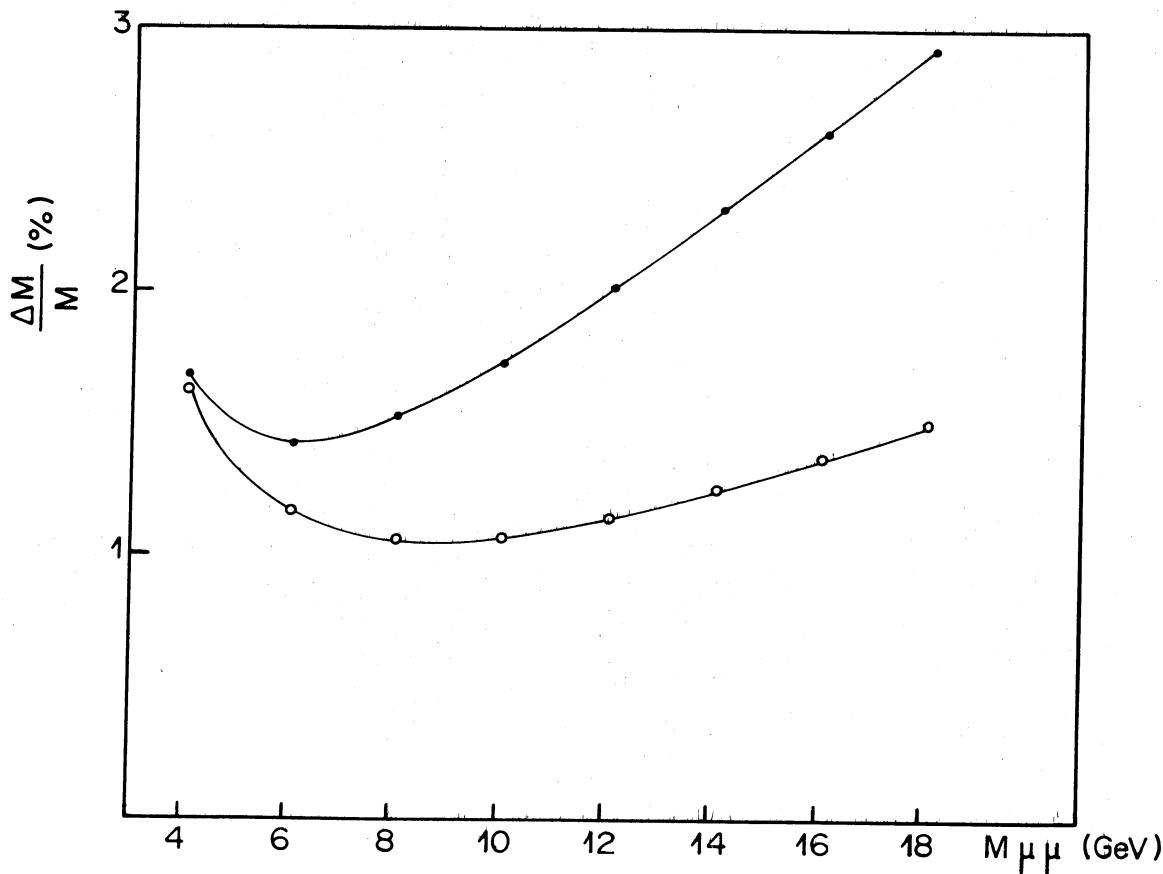


Fig.II.2 : Résolution en masse

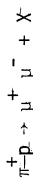
- (1) Dispositif normal
- (2) Dispositif à haute résolution

(iii) la détermination du rapport des sections efficaces π^-/π^+ de production de dimuons massifs. Alors que les faisceaux de π^- sont peu contaminés par d'autres particules, les faisceaux de π^+ contiennent 4 à 5 fois plus de protons que de pions. Cependant, comme le rapport de sections efficaces de Drell-Yan π^+/π^- augmente très rapidement avec la masse, presque tous les dimuons tels que $M^2/s > 0.2$ (ce qui correspond à $M \geq 9 \text{ GeV}/c^2$ à $p_{\text{lab}} = 200 \text{ GeV}/c$) proviennent de π^+ et leur section efficace peut être mesurée malgré la contamination des protons (remarquons également que les sections efficaces pour des protons sont connues et peuvent être soustraies). Dans cette zone de M^2/s les quarks de valence dominent ; on prévoit un rapport de sections efficaces π^-/π^+ (pour des cibles isoscalaires) égal à 4/1.

(iv) l'étude de la production de résonances avec une résolution en masse de l'ordre de 2 % et une sensibilité de quelques événements par jour pour des sections efficaces inférieures d'un ou deux ordres de grandeur à celle attendue par la production du π ($9.5 \text{ GeV}/c^2$). De plus la production de résonances telles que le π par des pions est d'un grand intérêt.

(v) l'étude de la dépendance de la section efficace en fonction de A (nombre de nucléons des noyaux du matériau cible)

Conclusion : La physique investie dans le cadre de cette expérience ne nécessite pas une étude exclusive de la réaction



En effet les prédictions théoriques qu'elle doit permettre de tester ne requièrent que la connaissance des valeurs prises par les variables dynamiques (énergie et impulsion) et la charge de chacun des deux muons. Il n'est donc pas nécessaire de détecter les hadrons provenant de la fragmentation des protons qui n'ont pas participé à l'annihilation. Cette situation présente deux avantages :

- le nombre et la complexité des détecteurs sont réduits
- il est permis d'absorber les hadrons à l'aide d'un absorbeur placé entre la cible et les détecteurs. Pour cette raison l'appareillage peut fonctionner avec un flux de particules arrosant la cible allant jusqu'à 10^{10} pions par pulse, sans pour autant que les détecteurs soient saturés (l'absorbeur atténuant et filtrant le flux de particules derrière la cible).

L'expérience NA 10 disposera de haut flux de particules incidentes permettant de vérifier les modèles théoriques avec une haute statistique, ce qui est nécessaire si l'on veut étudier un point particulier avec précision (une région cinématique particulière, par exemple, ou les grands moments transverses du dimuon ou les faibles valeurs de X où les quarks de la mer interviennent).

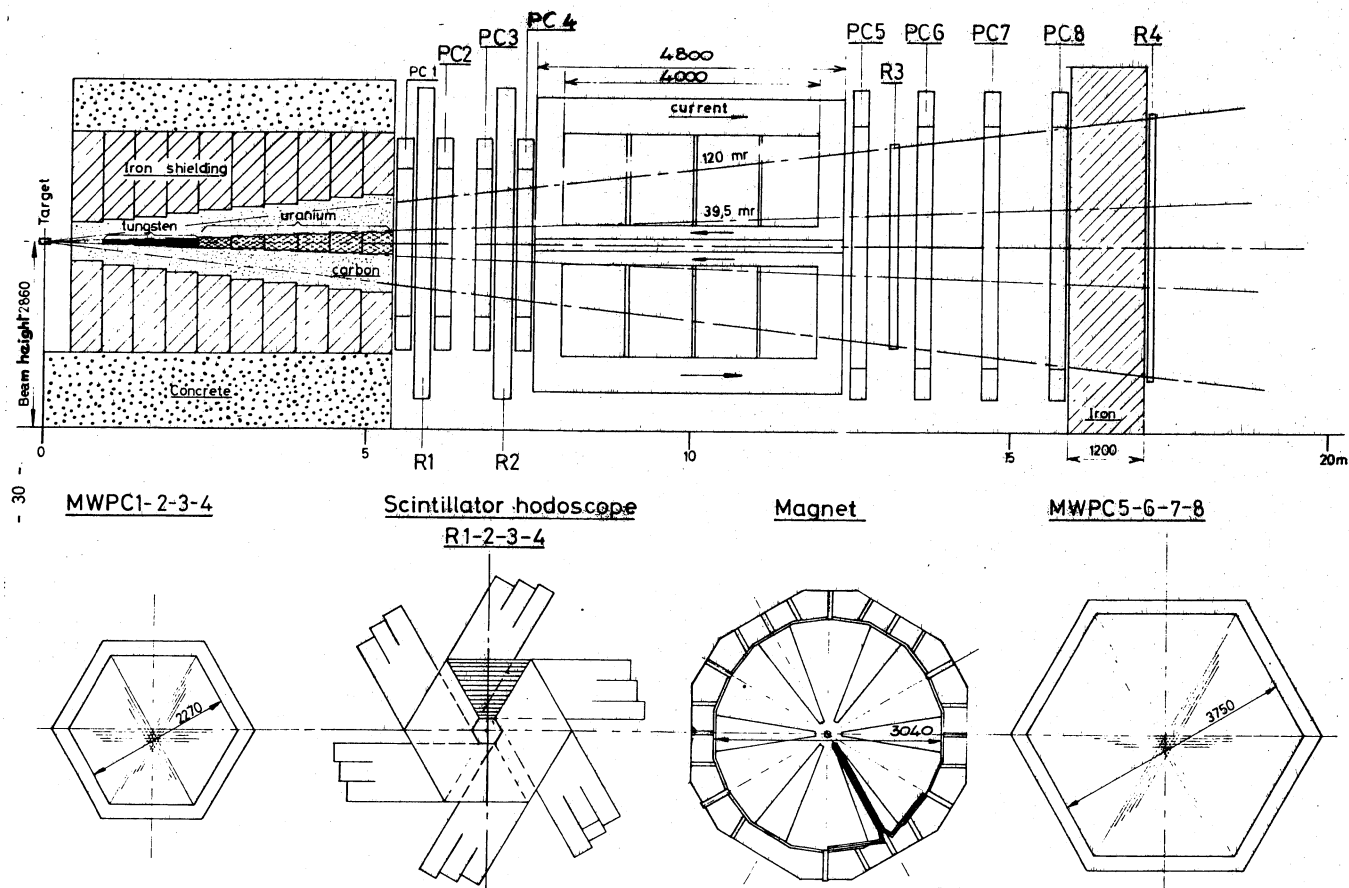


Fig. II.3 : Dispositif expérimental

B. DESCRIPTION DE L'APPAREILLAGE

Comme l'indique la figure II.3 l'appareillage se compose essentiellement des éléments suivants :

- i) une cible constituée d'un barreau de tungstène de 60 mm de long, ou d'un barreau de 90 mm de cuivre ou encore d'un barreau de 300 mm de carbone
 Note : il est prévu d'utiliser également une cible d'hydrogène permettant d'étudier les interactions sur protons libres
- ii) un absorbeur long de 4800 mm constitué d'une partie centrale en tungstène destinée à tuer le faisceau et les hadrons à petit angle, partie entourée d'un bloc de graphite destiné à tuer les hadrons à grand angle. Il est prévu de remplacer la partie périphérique en graphite par une portion de 4000 mm de graphite suivi de 800 mm de fer réalisant le meilleur compromis entre la longueur d'interaction et la longueur de radiation c'est à dire entre l'absorption de hadrons à grand angle et la précision sur la direction des muons, donc sur la masse des dimuons
- iii) un électro-aimant toroïdal, pulsé, constitué de six sextants égaux pointant sur l'axe du faisceau (qui est donc aussi celui de symétrie de l'aimant) et d'un noyau d'air (afin d'y réduire la diffusion multiple). Cet aimant crée un champ radial relativement uniforme en son centre, ce qui implique que l'angle de déflexion d'une particule chargée soumise à ce champ est inversement proportionnelle au moment transverse de la particule.
- iv) quatre hodoscopes (R_1, R_2, R_3, R_4) hexagonaux, orthogonaux à l'axe du faisceau, chacun constitué de six sextants égaux. Chaque sextant est constitué de 32 compteurs à scintillation perpendiculaires à la bissectrice du sextant ; deux hodoscopes se trouvent devant l'aimant : R_1 et R_2 , deux derrière : R_3 et R_4
- v) huit chambres proportionnelles multifils hexagonales (quatre à l'avant PC1 à PC4 ; quatre à l'arrière, PC5 à PC8 - de l'aimant) chacune d'entre elles se composant de trois plans de fils perpendiculaires à l'axe du faisceau. Le premier plan (à partir de la cible) est fait de fils parallèles à la direction verticale x, le second de fils parallèles à la direction u (qui fait un angle de $+60^\circ$ avec la verticale), le troisième de fils parallèles à la direction v (qui fait un angle -60° avec la verticale).

$P_T(\gamma)$	Acceptances	Type de coupure						
		PT > 1	PT > 1.25	PT > 1.5	RMC > 4	RMC > 4 PT > 1	RMC > 4 PT > 1.25	RMC > 4 PT > 1.5
3	A(M < 4 GeV)	0.75	0.58	0.36	0.08	0.07	0.05	0.03
	A(M > 4 GeV)	1	0.98	0.94	0.98	0.98	0.97	0.93
2	A(M < 4 GeV)	0.74	0.5	0.26	0.06	0.06	0.05	0.03
	A(M > 4 GeV)	1	0.99	0.97	0.98	0.98	0.98	0.96
1	A(M < 4 GeV)	0.8	0.56	0.26	0.08	0.08	0.08	0.07
	A(M > 4 GeV)	1	1	0.99	0.99	0.99	0.99	0.98
0.5	A(M < 4 GeV)	0.88	0.67	0.36	0.08	0.08	0.08	0.08
	A(M > 4 GeV)	1	1	1	0.98	0.98	0.98	0.98
0	A(M < 4 GeV)	0.96	0.76	0.47	0.05	0.05	0.05	0.05
	A(M > 4 GeV)	1	1	1	0.99	0.99	0.99	0.99

Tableau II.1 : Effet de coupures sur l'acceptance normalisée à l'acceptance géométrique

Remarque : Les dimensions des scintillateurs de R_1 sont en progression géométrique afin que chaque compteur voie la même longueur de cible (quel que soit l'angle). Les dimensions des sextants et des scintillateurs de R_2 se déduisent des éléments correspondants de R_1 par une homothétie ayant pour centre le milieu de la cible. Ceci signifie qu'une particule traversant le ième scintillateur du jème sextant de R_1 doit presque toujours traverser le ième scintillateur du jème sextant de R_2 (à la diffusion multiple dans l'absorbeur près), ce qui permet de réduire le taux de coïncidences fortuites.

C. SIMULATION DE L'EXPERIENCE

Dans le but d'estimer l'acceptance de l'appareillage et l'effet de diverses coupures sur l'impulsion transverse de muons et sur la masse invariante du dimuon nous avons utilisé un programme du type Monte Carlo qui génère des événements du genre étudié dans l'expérience

$$\pi^- p \rightarrow \gamma_V + X \rightarrow \mu^+ \mu^-$$

1) Paramètres du programme

La géométrie utilisée est celle du dispositif expérimental pour des pions incidents d'impulsion $p = 225 \text{ GeV}/c$

La paramétrisation utilisée pour reproduire les distributions en X_F et p_t est la suivante :

$$\frac{d\sigma}{dX_F dp_t} = (1 - |X_F|^2) p_t \exp(-1.5 p_t)$$

Le point d'interaction est généré uniformément en supposant une cible longue de 60 mm.

La distribution angulaire des muons est supposée isotrope (désintégration du photon virtuel dans le centre de masse).

On tient compte de la diffusion multiple des particules dans un absorbeur constitué uniquement de carbone ; l'angle de diffusion, en radians est donné par la relation suivante :

$$\Delta\theta = \frac{15}{p \cdot v} \sqrt{\frac{L}{\lambda_{\text{rad}}}}$$

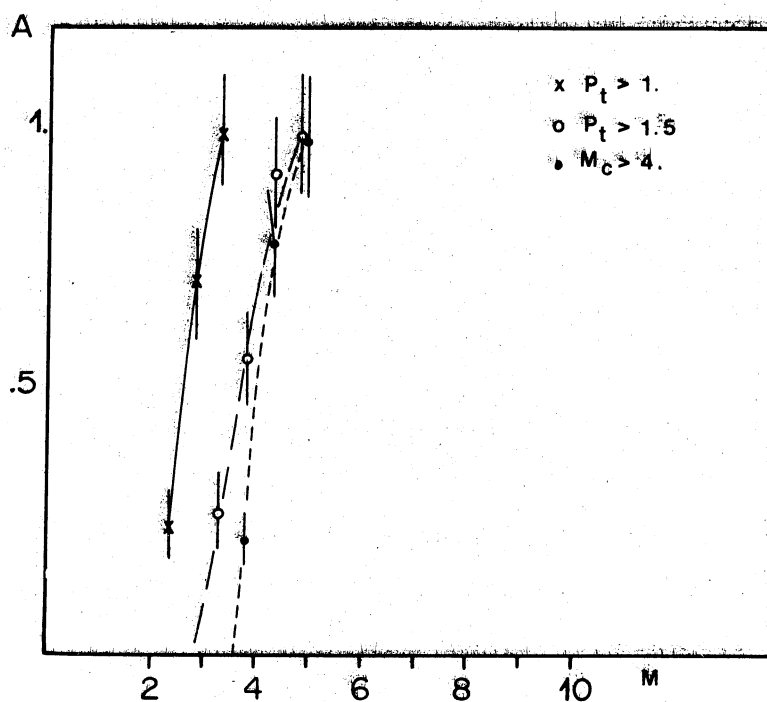


Fig.II.5 : Effet de coupures sur l'acceptance $P_T(\gamma) = 1. \text{ GeV/c}$

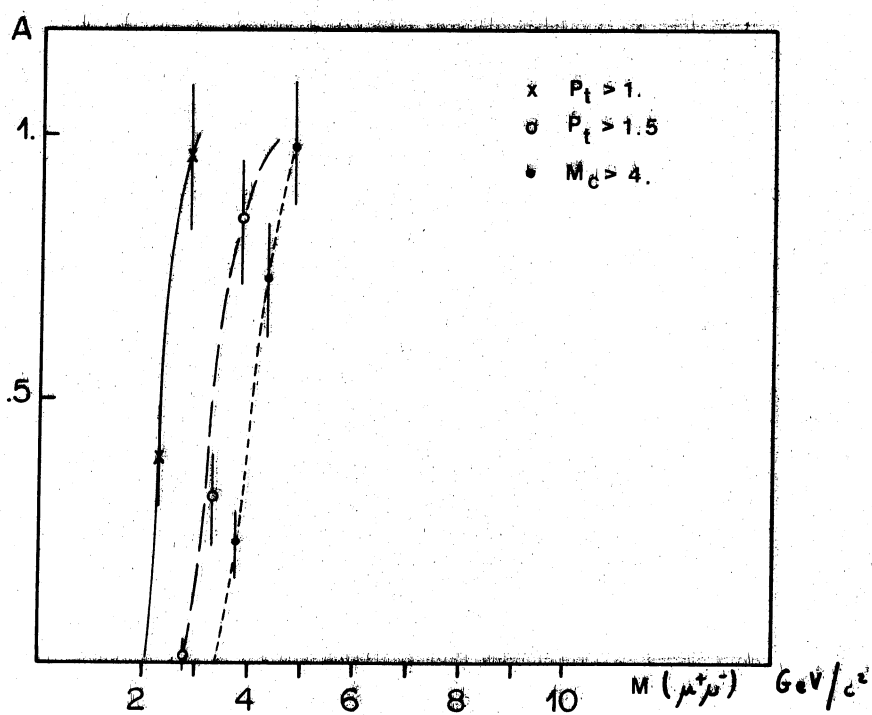


Fig.II.4 : Effet de coupures sur l'acceptance $P_T(\gamma) = 0.5 \text{ GeV/c}$

où : p et v sont respectivement le moment et la vitesse de la particule diffusée
 L est la longueur de l'absorbeur

L_{rad} est la longueur de radiation du matériau diffuseur

La déflexion d'une particule chargée lors de son passage dans l'aimant est approximée par la relation suivante :

$$D = \frac{0.066}{P_T} \text{ (en radian)}$$

où P_T est l'impulsion transverse de la particule.

2) Acceptances

L'acceptance géométrique de l'appareillage est donnée sur la figure II.1

L'expérience s'intéresse aux événements du type dimuon de haute masse ($M > 4 \text{ GeV}/c^2$), masse qui peut être approximée par la relation suivante (dans le cas où l'impulsion transverse du dimuon est nulle)

$$M = P_{T1} + P_{T2} \quad (1)$$

où P_{Ti} est l'impulsion transverse du muon i . Cette relation indique la possibilité de faire des coupures sur les faibles masses en sélectionnant les grandes impulsions transverses.

La formule exacte est :

$$M^2 = 2P_{T1}P_{T2}(1 - \cos \theta_{12}) \quad (2)$$

où P_i est l'impulsion du muon i et θ_{12} l'angle entre les deux traces, qui s'exprime par :

$$\theta_{12} = \cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos (\psi_1 - \psi_2)$$

où θ_i , ψ_i sont les angles polaire et azimutal d'un muon.

Les figures II.4 à II.7 et le tableau II.1 présentent l'effet de différentes coupures sur l'acceptance. Il apparaît clairement qu'à grande impulsion transverse du dimuon une coupure sur l'impulsion transverse des muons nous fait perdre des masses jusqu'à $6 \text{ GeV}/c^2$. Il est donc nécessaire de faire un calcul de masse en vue de sélectionner les événements.

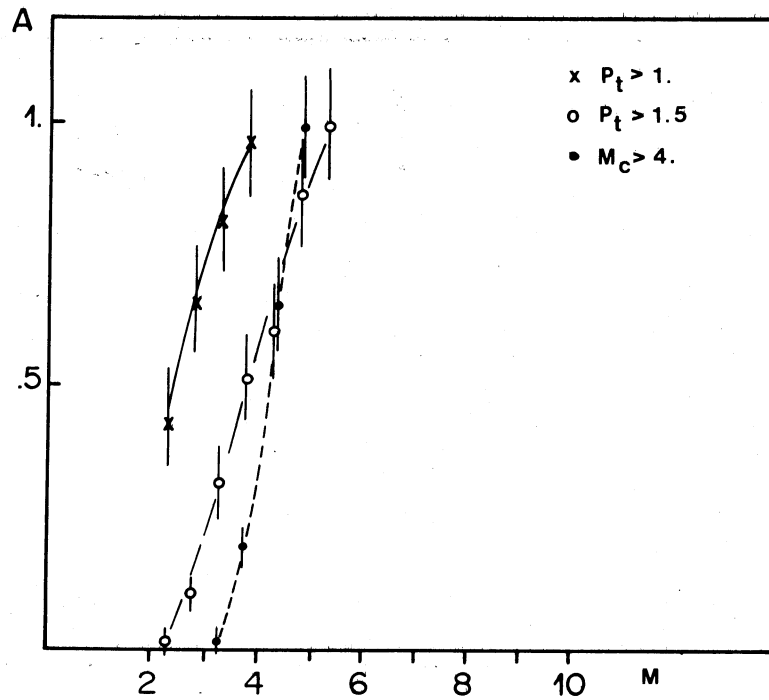


Fig.II.6 : Effet de coupures sur l'acceptance $P_T(\gamma) = 2. \text{ GeV}/c$

D. CRITERES DE FILTRAGE

Il apparaît clairement que le calcul de la masse est une très bonne façon d'approcher la sélection des événements.

De plus l'examen des expressions approchée (1) et exacte (2) de la masse du dilepton, confirmé par la simulation programmée, nous a permis de dégager les critères supplémentaires suivants :

- i) élimination des événements dont les deux traces sont dans le même sextant ou dans deux sextants adjacents
- ii) recherche, dans un sextant, de la trace de plus grand moment transverse
- iii) dans le cas où il y a ambiguïté pour l'application du critère ii) on conserve, à impulsion transverse égale, la trace d'angle polaire θ le plus faible.

II. LE FILTRAGE EN LIGNE : CAS GENERAL

A. INTRODUCTION

La détection électronique actuelle permet l'acquisition de taux élevés d'événements, taux qui vont encore augmenter dans les expériences futures. La complexité des appareillages et des critères de sélection s'accroît également. Les systèmes de déclenchement rapide (fast trigger) ne peuvent entièrement satisfaire les expérimentateurs : leur rigidité leur interdit d'être trop spécialisés dès l'abord. Les événements intéressants sont en général contaminés par une multitude d'inintéressants. Pour distinguer les bons événements du bruit de fond on fait appel à des techniques logicielles.

Normalement cette sélection logicielle est faite dans la première

phase de l'analyse des événements, analyse faite sur gros ordinateur dans un centre de calcul. Ceci implique cependant que beaucoup de mauvais événements auront été enregistrés pendant l'expérience en vue d'analyse ultérieure. Les effets suivants en résultent :

- les temps-morts dus à l'enregistrement en ligne de mauvais événements
- le grand nombre de bandes magnétiques qui doivent être écrites, puis traitées ce qui alourdit et retarde l'exploitation des résultats.
- une augmentation considérable du temps de calcul (coûteux) sur grosse machine

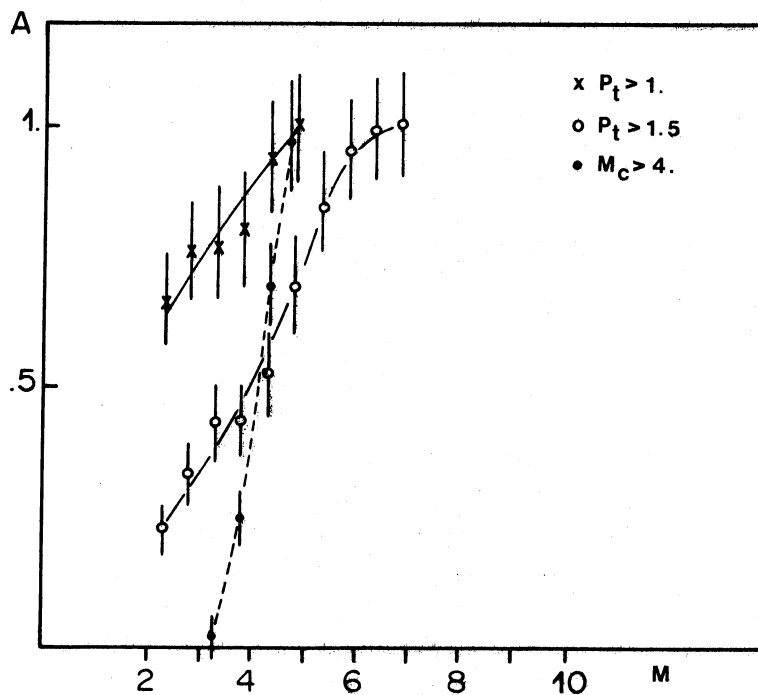


Fig. II.7 : Effet de coupures sur l'acceptance $P_T(\gamma) = 3.$ GeV/c

Chacun de ces effets pris séparément constitue une raison suffisante pour essayer de sélectionner les données à un stade antérieur : avant que les événements ne soient enregistrés sur bande magnétique. C'est ce qu'on entend par "filtrage en ligne" (on-line filtering).

B. LE FILTRAGE EN LIGNE

1) Le volume des données

L'ensemble de l'appareillage électronique de détection est la source d'un nombre très important de données, de l'ordre du millier de mots par événement. Ces données peuvent être :

- des coordonnées
- des hauteurs d'impulsions
- des configurations de compteurs touchés
- des informations générales (courants dans les aimants, par exemple)

2) Limitation de la sélectivité du déclenchement

La première étape (et la plus rapide) dans l'électronique de déclenchement est basée sur les signaux des compteurs à scintillation. Cette logique très rapide ne peut tenir compte de toute l'information additionnelle, ce qui limite sa sélectivité. Diverses raisons rendent l'amélioration de cette sélectivité difficile. Pour distinguer les bons événements du bruit de fond des détecteurs, la corrélation en temps des particules est essentielle. La résolution en temps des circuits de coïncidence est de l'ordre de la nanoseconde ou de quelques nanosecondes, donc l'électronique doit être très rapide. Les modules fonctionnels utilisés s'accommodent seulement d'un nombre faible de fonctions logiques simples avec un petit nombre d'entrées dans chaque module. Pour un grand nombre de compteurs et des critères de déclenchement compliqués le nombre de modules requis peut aller au-delà des limites raisonnables.

Les compteurs utilisés dans le trigger sont en général assez grands, ce qui implique une limitation de la sélectivité basée sur les quantités cinématiques.

Une autre limitation sur le trigger rapide provient du fait que les données très précises des chambres à fils ne peuvent être utilisées, soit parce qu'elles représentent un volume trop important, soit parce qu'elles ne sont pas disponibles à l'instant voulu.

3) Le bruit de fond

Le trigger rapide, essentiel pour la détection des corrélations en temps des particules et pour donner le signal de départ des actions suivantes, laisse l'expérimentateur en face d'événements hautement contaminés par le bruit de fond. Celui-ci est de nature variée, par exemple :

- des coïncidences fortuites de signaux
- des événements satisfaisant le trigger mais qui ne sont pas du type recherché
- des événements du type recherché mais qui ne sont pas dans la zone cinématique qui intéresse l'expérimentateur

4) Conclusion

Il est nécessaire d'appliquer des critères de sélection en temps réel plus sévères que le trigger rapide (avant que l'événement ne soit écrit sur bande magnétique en vue d'analyse off-line). Ce filtrage en ligne peut se faire en temps réel à plusieurs niveaux de décision différents : chaque niveau agissant sur les événements acceptés par le niveau précédent. Ce filtrage est évidemment toujours plus lent que la logique rapide de déclenchement, mais le plus souvent il doit s'effectuer sous des contraintes en temps très drastiques pour éviter :

- les temps morts
- la nécessité de travailler à des taux de déclenchement plus faibles

Pour favoriser un filtrage efficace il est utile de se souvenir des principes de base suivants : il faut :

- disposer d'un système de lecture rapide (de l'ordre de 100 nanosecondes par mot)
- faire une lecture sélective, rendant possible le traitement des données de certains détecteurs seulement ou permettant de lire d'abord les détecteurs nécessaires à la décision

- faire une lecture parallèle : chose importante si on peut effectuer un traitement significatif sur des données partielles
- disposer de données bien structurées pour éviter au maximum la perte de temps due au formatage éventuel avant traitement
- pouvoir adapter le filtre à l'évolution de l'expérience, donc rester évolutif, programmable.

C. FILTRES LOGICIELS (SOFTWARE)

1) Méthodes logicielles et matérielles (hardware)

Il y a quelques années le filtrage en ligne de type logiciel résidait nécessairement dans l'utilisation de mini-ordinateurs. Les méthodes logicielles étaient avantageuses par la flexibilité qu'elles offraient, mais très lentes comparées aux méthodes hardware. Les progrès technologiques, par exemple, l'introduction des microprocesseurs par tranches de bits (bit-slice), tendent à rapprocher les deux méthodes. Les "bit-slice" sont des blocs assemblables à partir desquels on peut construire des processeurs programmables, tout en permettant une grande liberté dans la structure et les fonctions du processeur. Ces processeurs allient la flexibilité à la vitesse quand ils sont utilisés à bon escient ; la vitesse maximale est obtenue lorsque le processeur est étroitement intégré à l'expérience, directement interfacé au système de lecture des différents détecteurs.

Un bit-slice peut être programmé directement en micro-code par lequel on connaît le détail du hardware : le processeur est encore "programmable" mais pas de façon triviale ; on peut remédier à cet inconvénient en faisant ressembler le produit final à un autre ordinateur. Cette émulation peut aller jusqu'à produire une machine capable d'exécuter un programme originellement écrit pour une autre machine. Ceci fait gagner en maniabilité et en flexibilité surtout si les programmes peuvent être écrits en langage évolué ; par contre on perd en vitesse d'exécution.

On peut difficilement faire des comparaisons de vitesse entre un filtre hardware et un filtre software en termes généraux, car elle dépend étroitement du problème particulier auquel on est confronté, de la conception générale du système et, pour le logiciel, de la machine et du langage utilisés. Disons cependant qu'en gros un miniordinateur ne sera pas capable en général de traiter plus de 10 à 50 K mots par seconde ; un bit-slice pourra en faire 2 à 5 fois plus en mode programmé, et 10 à 20 fois plus en mode microprogrammé, principalement grâce à l'utilisation de mémoire rapide. Le prix de revient

"relativement" faible d'un bit-slice incite à en utiliser plusieurs pour le filtrage : une bonne vitesse peut être obtenue en faisant du "multiprocessing". Mais même cette technique permet difficilement d'égaliser des filtres câblés.

Pour ce qui est de la flexibilité, celle-ci diminue quand on passe du miniordinateur au bit-slice et de celui-ci à un processeur câblé. La flexibilité est proportionnelle à la capacité d'un système à se prêter au changement, à des tâches diverses.

De façon générale le dilemme entre vitesse et flexibilité sera résolu par les choix suivants :

- L'option processeur câblé lorsque la vitesse est un facteur primordial et quand le problème est bien connu jusque dans les détails pour permettre d'allouer le temps nécessaire à la conception et la réalisation du hardware
- L'option logique si la vitesse n'est pas un facteur essentiel et si l'on s'attend à devoir ajuster souvent, voire changer, les algorithmes de filtrage
- le microprocesseur par tranches si l'on voit clairement la façon de l'intégrer à l'environnement expérimental mais si l'on ne connaît pas exactement l'algorithme de filtrage à exécuter. L'effort total requis (conception, construction, développement d'outils logiciels tels que cross-assembleur, software utilisateurs) est très supérieur à celui nécessaire à l'élaboration d'un processeur câblé sophistiqué.

2) Amélioration des performances du logiciel.

Il est possible d'augmenter la vitesse du logiciel sans perdre la flexibilité : voici quelques méthodes, la première étant triviale.

a) Programmation efficace

Cela signifie : obtenir la meilleure vitesse d'exécution des programmes, chose qui entre en conflit généralement avec l'intelligibilité, la lisibilité des programmes. L'utilisation du langage assembleur permet d'améliorer les performances.

b) Microprogrammation

Certains miniordinateurs sont micro-programmables par l'utilisateur, mais cette "facilité" est relativement peu utilisée, car il est considérablement

plus difficile de micro-programmer un ordinateur que d'écrire un programme en langage évolué ou même en langage assembleur. Cependant un facteur 5 à 10 peut être gagné de cette façon, en évitant de stocker des données intermédiaires ou en utilisant la possibilité d'effectuer certaines opérations élémentaires en parallèle (simultanément) avec d'autres et par suppression des accès en mémoire centrale. Le gain est surtout important pour des opérations répétitives nécessitant un ou deux registres pour les résultats intermédiaires (algorithme de multiplication par exemple).

c) Adjonction d'opérateurs câblés

C'est une technique bien connue : elle consiste à adjoindre à l'ordinateur des modules fonctionnels câblés qui effectuent à très grande vitesse des opérations spécialisées, mais il y a des difficultés : les entrées-sorties ralentissent les opérations.

d) Multiprocessing

Deux approches différentes peuvent être envisagées lorsqu'on veut répartir une tâche entre plusieurs processeurs.

- le processing en parallèle sur plusieurs événements en même temps.

Du point de vue conceptuel, c'est très simple, un nouvel événement est alloué à un processeur quand il a fini avec le traitement en cours. Cependant des problèmes se posent : il faut stocker temporairement toutes les données correspondant aux événements en cours de traitement. Si ceci n'est pas possible, on ne peut rien gagner par rapport au multiprocessing

- distribuer le traitement d'un seul événement à plusieurs machines, chacune effectuant une partie seulement de la tâche globale. Ceci est séduisant dans son principe, mais difficilement réalisable en pratique. Diviser une tâche en sous-tâches quasi-indépendantes n'est pas toujours chose facile, mais parfois c'est possible, auquel cas il reste encore la synchronisation des tâches et la communication des données, des résultats et du status de chaque processeur à assurer, toutes choses assez délicates à réaliser.

e) "Self-made" processeur

La dernière possibilité d'obtenir un filtre logiciel rapide consiste à réaliser soi-même une machine rapide programmable. Du point de vue hardware cette solution est séduisante car on peut concevoir et réaliser une machine dont la structure est spécifiquement adaptée au problème à résoudre, et

lui adjoindre des unités fonctionnelles spécialisées. De cette façon on arrive à dépasser largement les performances d'un mini-ordinateur. Mais d'un point de vue software, l'enthousiasme risque d'être mitigé car la programmation d'un tel processeur sera inévitablement assez difficile. Pour la rendre plus aisée il faut faire un effort important consistant à développer des programmes d'assemblage, des chargeurs, linkers, etc....

D. LES MICROPROCESSEURS PAR TRANCHES

Le microprocesseur par tranches forme une sorte de transition entre les filtres software et hardware. Il est une manière commode de disposer sur un seul circuit intégré des unités arithmétiques et logiques ainsi qu'un ensemble de registres fonctionnant sur un nombre de bits réduit (2 ou 4 en général). Ils n'apportent pas d'idées architectoniques fondamentalement nouvelles, mais il est particulièrement aisé de construire un processeur avec des bit-slice. Il suffit en effet de câbler le nombre voulu, n , en parallèle, pour obtenir un processeur travaillant sur une longueur de mot voulue ($2n$ ou $4n$ en général). Si on se donne la peine de développer un software utilisateur qui rend la programmation de la machine "aisée", celle-ci n'est plus fondamentalement différente d'un miniprocesseur ; elle est particulièrement indiquée dans un contexte tel que le filtrage de données, où il n'est pas nécessaire de disposer de périphériques sophistiqués ni d'exécuter d'énormes programmes.

E. FILTRES CABLES

L'avantage saillant d'un processeur hardware est le suivant : pour un algorithme donné, on peut réaliser la structure la plus performante possible, à l'intérieur des contraintes budgétaires et technologiques. Pour atteindre la vitesse optimale il y a un prix à payer : un temps long pour le développement d'un matériel assez rigide, peu flexible et coûteux, souvent de maintenance difficile.

III. APPLICATION A L'EXPERIENCE NA10 : DESCRIPTION DE LA STRUCTURE DE FILTRAGE ADOPTEE.

Sur l'expérience a été implanté un système de lecture rapide des détecteurs appelé RMH, celui-ci est capable de lire un événement à raison de 100 nanosecondes par mot de 16 bits, c'est à dire 100 microsecondes pour un événement de 1000 mots de 16 bits. Pour ne pas pénaliser le système de lecture rapide, l'ensemble de filtrage ne doit pas consommer plus de 100 microsecondes par événement. Les considérations du paragraphe précédent nous ont amenés à conclure que l'utilisation d'un microprocesseur "bit-slice" calculateur est particulièrement adapté à l'expérience : certains critères de filtrage étant définis et d'autres devant être facilement adaptables (multiplicité des traces, par exemple)

L'utilisation d'un calculateur de masse dans le filtrage des événements a été étudiée. Les conclusions suivantes ont été dégagées :

1) Précision des calculs

La formule précise (2) de la masse comporte des termes très petits du type $\sin \theta$ et des termes grands du type $\cos \theta$, $\cos \Delta \phi$ et P_i ; la simulation de ce calcul sur grosse machine nous a permis de mettre en évidence l'insuffisance d'un calcul sur 16 bits ; un calcul sur 24 bits est souhaitable.

2) Vitesse

La durée du calcul ne doit pas dépasser 100 microsecondes par événement. Ceci impose une structure multibus, des opérateurs arithmétiques câblés, un ou plusieurs C.P.U. parmi les plus rapides. L'ensemble doit être microprogrammable pour réduire au maximum les temps d'exécution, et programmable pour permettre des mises au point et des tests dans de bonnes conditions

3) Structure adoptée

a) ordinateurs

Le C.P.U. multibus le plus rapide est l'INTEL 3000 (150 nanosecondes de temps de cycle) qui permet de plus un maximum d'horizontalité en microprogrammation.

b) opérateurs câblés

*opérateurs arithmétiques

Des opérateurs sont indispensables pour :

- la multiplication entière et flottante
- la conversion entier \rightarrow flottant

- la conversion flottant \rightarrow entier
- la réduction au même exposant nécessaire en vue d'addition de nombres flottants

Ces opérations sont réalisables actuellement en 200 nanosecondes chacune.

*recherche du plus grand moment transverse

Cette recherche ne peut être faite qu'en mode séquentiel et nécessite par conséquent 32 cycles de base par sextant. Le temps d'exécution d'une telle opération est d'environ 5 microsecondes. Pour ne pas pénaliser l'ensemble, 6 opérateurs (un par sextant) seront activés simultanément.

Chaque opérateur fournit, dans un sextant, le numéro, V_i , de la coïncidence $R_{1i}R_{2i}$ retenue, la valeur du plus grand moment transverse, le numéro du sextant et la multiplicité des traces.

c) durée de l'opération

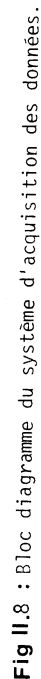
Dans ces conditions (supposant de plus que tous les opérandes sont tabulés en mémoire) un calcul de masse demande environ 20 microsecondes (comprisant la lecture de deux opérateurs de recherche du plus grand moment transverse, la lecture en mémoire des opérandes : $\sin \theta_i$, $\cos \theta_i$, ..., les tests de signe, les multiplications et additions et les tests de débordement). Dans le cas le plus défavorable (lorsque tous les sextants sont touchés) 15 calculs de masse sont nécessaires pour le même événement : la limite supérieure de 100 microsecondes par événement demande donc quatre ensembles complets (CPU + module multiplieur) fonctionnant simultanément (compte tenu d'une marge de sécurité d'une vingtaine de microsecondes).

d) description de l'ensemble de filtrage

L'ensemble de filtrage est illustré par la figure II.8. On y

distingue schématisés :

- l'ensemble de détection
- la matrice de coïncidence sur laquelle viennent se greffer les opérateurs de recherche du plus grand moment transverse
- le système de lecture rapide RMH
- les quatre calculateurs de masse : "GESPRO" (chacun constitué d'un CPU et d'un opérateur "multiplieur") connectés à
- l' "EVENT BUFFER" composé de 4 mémoires tampon contenant chacune un événement complet
- le contrôleur (CTRL) qui alloue les différents événements aux calculateurs et qui autorise le transfert vers :



Ce qui signifie que le système de filtrage doit rejeter 9 événements sur 10 pour ne pas bloquer l'acquisition.

IV LA MICROMACCHINE DE CORDO 16)

A. LA PARTIE OPERATIVE DE LA MICROMACHINE

Elle est construite autour d'une unité de traitement opérant sur des mots de 24 bits et utilisant les circuits INTEL-3002 CPE microprocesseurs en tranches de 2 bits

1. Description de l'unité CPE

Les différents éléments constitutifs d'un CPE (Control Processing

Element), c'est à dire d'une tranche d'unité de traitement opérant sur 2 bits sont les suivants : (voir fig. II.9)

a) l'opérateur arithmétique et logique

Il constitue le centre de la partie opérative : il est capable

d'effectuer les opérations arithmétiques et logiques classiques ; l'ensemble des opérations est donné dans le tableau II.2

Cet opérateur s'intègre dans une structure de type "trois registres" comme l'indiquent la figure II.7 et le paragraphe suivant :

b) les registres

Ils se répartissent en 3 catégories :

- les registres généraux : R_0 à R_9 et T , dont un seul peut être en entrée ou en sortie dans une opération
- le registre spécialisé d'adresse de la mémoire : MAR
- le registre accumulation AC ; il peut apparaître à la fois en entrée et en sortie de l'opérateur et sert à charger le bus de données vers la mémoire (M)

F.GROUP	R.GROUP	MICRO-FUNCTION
0	I	$R_n + (AC \wedge K) + CI \rightarrow R_n$, AC
	II	$M + (AC \wedge K) + CI \rightarrow AT$
	III	$AT_L \wedge (I_L \wedge K_L) \rightarrow RO$, $LI \vee (I_H \wedge K_H) \wedge AT_H \rightarrow AT_H$ $(AT_L \wedge I_L \wedge K_L) \vee (AT_H \wedge I_H \wedge K_H) \rightarrow AT_L$
1	I	$K \vee R_n \rightarrow MAR$, $R_n + CL + K \rightarrow R_n$
	II	$K \vee M \rightarrow MAR$, $M + CI + K \rightarrow AT$
	III	$(AT \vee K) + (AT \vee K) + CI \rightarrow AT$
2	I	$(AC \wedge K) - 1 + CI \rightarrow R_n$ (see Note 1)
	II	$(AC \wedge K) - 1 + CI \rightarrow AT$
	III	$(I \wedge K) - 1 + CI \rightarrow AT$
3	I	$R_n + (AC \wedge K) + CI \rightarrow R_n$
	II	$M + (AC \wedge K) + CI \rightarrow AT$
	III	$AT + (I \wedge K) + CI \rightarrow AT$
4	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$, $R_n \wedge (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$, $M \wedge (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$, $AT \wedge (I \wedge K) \rightarrow AT$
5	I	$CI \vee (R_n \wedge K) \rightarrow CO$, $K \wedge R_n \rightarrow R_n$
	II	$CI \vee (M \wedge K) \rightarrow CO$, $K \wedge M \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$, $K \wedge AT \rightarrow AT$
6	I	$CI \vee (AC \wedge K) \rightarrow CO$, $R_n \vee (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (AC \wedge K) \rightarrow CO$, $M \vee (AC \wedge K) \rightarrow AT$
	III	$CI \vee (I \wedge K) \rightarrow CO$, $AT \vee (I \wedge K) \rightarrow AT$
7	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$, $R_n \oplus (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$, $M \oplus (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$, $AT \oplus (I \wedge K) \rightarrow AT$

NOTE:
1. 2's complement arithmetic adds 111...11 to perform subtraction of 000...01.

SYMBOL	MEANING
I, K, M	Data on the I, K, and M buses, respectively
R_n	Contents of register n (R-Group I)
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
CI	Data on the carry input
CO	Data on the carry output
L, H	As subscripts, designate low and high order bit, respectively
+	2's complement addition
-	2's complement subtraction
\wedge	Logical AND
\vee	Logical OR
\oplus	Exclusive-NOR
\rightarrow	Deposit into

Tableau n° II.2 : Liste des micro-opérations du C.P.E. INTEL 3002.

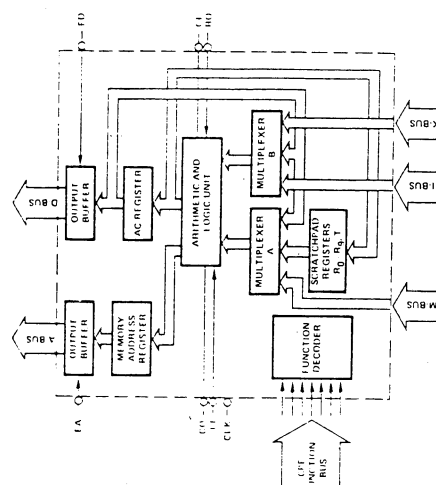


Figure n° II.9 : Le C.P.E. INTEL 3002.

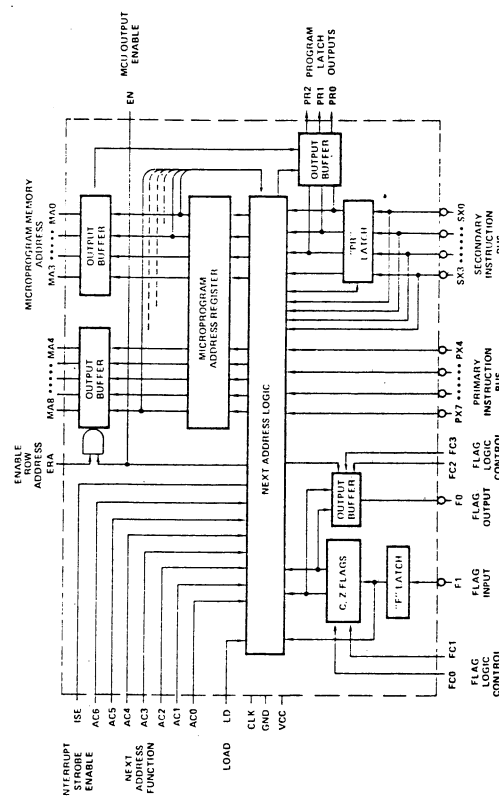


Figure n° II.10 : Le circuit M.C.U. INTEL 3001.

1. L'unité de contrôle

Elle est composée du circuit INTEL-3001 MCU (Microprogram Control Unit). La structure est donnée sur la figure II.10. Cet organe a pour rôle de calculer l'adresse (dans la mémoire de commande) de la micro-instruction suivante à exécuter. Cette fonction est effectuée par l'opérateur de calcul d'adresse en fonction :

- de l'adresse courante du microprogramme
- du contenu de la fonction de commande d'adresse
- du contenu du latch F (indicateur d'état du report de l'unité centrale ou de la logique de décalage FI)
- du contenu des "drapeaux" (flags) C et Z qui sont des indicateurs ayant un caractère permanent, que l'on peut commander grâce aux deux fonctions (bits) de contrôle FC_0 et FC_1 permettant de les forcer à la valeur 0, 1, ou du latch F
- de la valeur du bus PX véhiculant les 4 bits de poids fort du code opération de l'instruction machine en cours (en provenance de la mémoire centrale)
- du contenu du bus SX contenant les 4 bits de poids faible du même code opération.

La capacité d'adressage est de 512 micro-instructions (repérées grâce à 9 bits) organisées dans une représentation matricielle en lignes et colonnes

- 32 lignes numérotées de 0 à 31, soit 5 bits
- 16 colonnes numérotées de 0 à 15, soit 4 bits

Un dispositif matériel permet d'augmenter la capacité adressable en offrant la possibilité de mettre en parallèle plusieurs ensembles de 512 micro-instructions (ensembles baptisés "plans"). Deux plans sont nécessaires dans notre application.

Le nombre de plans de la mémoire de commande est invisible à l'organe de contrôle du séquençement qui adressera la mémoire de contrôle à l'aide d'un numéro de ligne et d'un numéro de colonne. Le dispositif matériel s'intercalant entre l'unité de contrôle et la mémoire complètera cette adresse par un numéro de plan.

Le MCU permet aussi de commander un indicateur F0 (en sortie vers l'unité de traitement)

2. La gestion du séquençement

La logique de séquençement du M.C.U. ne permet pas, à partir d'une microinstruction, d'atteindre n'importe quelle autre microinstruction de la mémoire de commande (à cause d'un adressage unitaire).

c) les indicateurs

Ils sont de trois types :

- les indicateurs de retenue
- les indicateurs de débordement
- les indicateurs de décalage

d) les multiplexeurs

Ils servent à sélectionner les entrées de l'opérateur

- le multiplexeur A choisit l'un des 11 registres généraux, les données du bus M en provenance de la mémoire et le registre AC
- le multiplexeur B sélectionne le registre AC ou les données du bus d'entrée I ou encore du bus K

e) la logique de décodage

Les commandes en provenance de la partie contrôle sont décodées :

- pour sélectionner la fonction à exécuter par l'opérateur
- pour générer le numéro du registre utilisé
- pour contrôler les multiplexeurs A et B

2. La commande de l'unité de traitement

Les fonctions arithmétiques et logiques réalisées par l'unité de traitement sont classées en 8 groupes (F-groupes).

Dans chaque groupe F les registres sont répartis en 3 classes (R-groupes) 7 bits sont nécessaires pour désigner toutes les fonctions réalisables (F_0 à F_6). Les bits de poids fort, F_6 à F_4 indiquent la fonction (F-groupe), les bits de poids faible, F_3 à F_0 , repèrent le registre concerné par l'opération. Le tableau II.2. résume cette situation.

B. LA PARTIE CONTRÔLE DE LA MICROMACHINE 17)

Elle est constituée de l'unité de contrôle et de la mémoire de commande. Présentons succinctement l'unité de contrôle, puis le séquençement et le contrôle des indicateurs.

Les trois types de sauts possibles sont les suivants :

- les sauts inconditionnels
- les sauts conditionnés par la valeur d'un indicateur
- les sauts conditionnés par le contenu du bus PX (4 bits) ou du registre-tampon PR (4 bits)

a) les sauts inconditionnels

- JCCn : saut dans la colonne courante, ligne n
- JCRn : saut dans la ligne courante, colonne n
- JZRN : saut à la microinstruction située à la ligne 0, colonne n
- JCCn : saut dans la colonne courante à la ligne x, x étant obtenu par concaténation du champ de la microinstruction et du numéro de la ligne courante

b) les sauts conditionnés par la valeur d'un indicateur

- JFL : saut selon la valeur du tampon F ; les deux adresses possibles ne diffèrent que pour le dernier bit du numéro de la colonne, bit qui aura la valeur de F
- JCF : saut selon la valeur de l'indicateur C, avec les mêmes règles que pour JFL
- JZF : saut selon la valeur de l'indicateur Z, avec les mêmes règles que précédemment

Un nombre limité de lignes, 16 (JFL) ou 8 (JCF, JZF) selon les sauts, sont accessibles par ces sauts. Les colonnes accessibles sont les colonnes (2,3 ou (10,11)) selon le numéro de la colonne de la microinstruction précédente.

c) les sauts d'éclatement selon les contenus du bus PX ou du tampon PR.

Le bus PX est chargé avec la partie haute, le registre-tampon PR avec la partie basse, du code opération de l'instruction en début d'exécution. Les sauts possibles sont les suivants :

- JPR : ce saut permet le déplacement dans les lignes et les colonnes : le nouveau numéro de ligne est choisi en fonction du numéro de la ligne courante et du champ de la microinstruction ; ce saut permet d'accéder à toutes les colonnes puisque le numéro de la prochaine colonne est donné par les 4 bits du registre-tampon PR

- JLL : saut en fonction de la partie gauche (2 bits) du registre-tampon PR ; la différence entre ce saut et le saut JPR est la restriction sur les colonnes accessibles : on ne peut atteindre que les colonnes de numéro compris entre 4 et 7
- JRL : saut en fonction de la partie droite (2 bits) de PR ; 4 lignes sont accessibles à proximité de la ligne courante, les colonnes accessibles auront un numéro compris entre 12 et 15
- JPX : saut en fonction du bus PX ; le numéro de la ligne atteinte sera obtenu en fonction du numéro de la ligne courante et du champ de la microinstruction, le numéro de la colonne sera formé des 4 bits du bus PX : toutes les colonnes sont accessibles.

3. Le contrôle des indicateurs (FC₀-FC₃).

Les fonctions FC₀-FC₃ servent à conserver la valeur de la retenue en provenance de l'unité de calcul (FI) et à contrôler la valeur de la retenue en entrée de l'unité de traitement (FO).

Le tableau II.3 donne les commandes des fonctions FC₀-FC₃ sur les indicateurs C et Z et sur la retenue FO en sortie du MCU.

FC ₀	FC ₁	Action
0	0	Positionne C et Z à la valeur de F (FI)
0	1	Positionne C à la valeur de F
1	0	Conditionne Z à la valeur de F
1	1	Conserve leur valeur à C et Z

FC ₃	FC ₂	Action
0	0	Force FO à 0
0	1	Force FO à la valeur de C
1	0	Force FO à la valeur de Z
1	1	Force FO à 1

Tableau II.3 : Champ de contrôle des indicateurs

CHAPITRE III

LOGICIEL

I. LES INSTRUCTIONS

A. Généralités

Une instruction est définie par son code opération. Celui-ci provoque l'exécution d'une séquence microprogrammée qui peut faire exécuter une opération câblée.

GESPRO^{11,12,13} est une machine de type intermédiaire puisque les opérateurs câblés adjoints à l'unité centrale définissent d'une part une séquence microprogrammée et d'autre part des séquences câblées. Ceci est possible du fait d'une structure spécifique à l'architecture de la micro-instruction. Le format de celle-ci est donné sur la figure III.1.

On distingue d'une part :

- une zone de contrôle d'adresse
- une zone de contrôle des indicateurs
- la commande de chargement du MCU à partir du bus mémoire
- les fonctions unité centrale (arithmétiques et logiques)
- le bus de masque K

Ceci constitue la zone nécessaire pour assurer le fonctionnement de l'infrastructure de base et caractérise le fonctionnement microprogrammé.

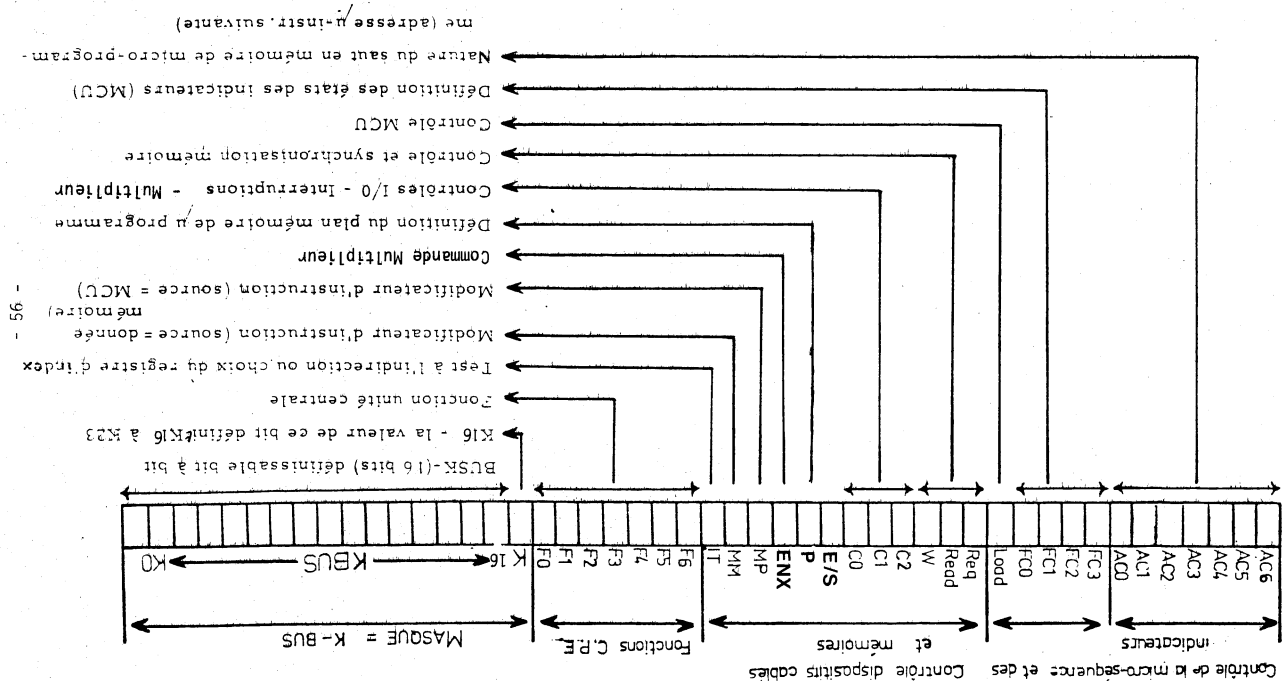


Fig. III.1. Format de la micro-instruction

D'autre part nous avons une zone de 12 bits permettant de définir le fonctionnement des opérateurs câblés. On y distingue :

- le contrôle de la mémoire centrale (propre et extension aux modules de recherche du plus grand moment transverse et à l'Event Buffer)
- la commande des Entrées/Sorties, du module d'interruptions et du module "multiplicateur"
- les modificateurs d'instructions
- le bit de changement de plan de microinstructions
- le bit de débranchement conditionnel à l'opération en cours (indirection, indexation, débordement...)

B. Format général d'une instruction

Les instructions GESPRO sont formées d'un ou plusieurs mots de 24 bits.

Le premier mot est formé :

- d'un champ code opération sur 8 bits
- d'un champ adresse ou donnée sur 16 bits

Les autres mots peuvent être formés de :

- données
- constantes
- adresses

sur un nombre quelconque de bits.

Une instruction peut être structurée en deux zones (fig.III.2)

- la zone commande, constituée par les 8 premiers bits
- la zone argument, constituée par le reste de l'instruction

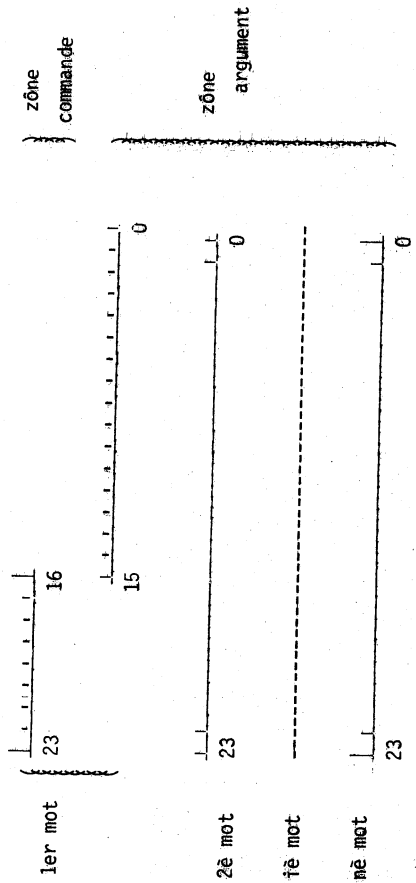
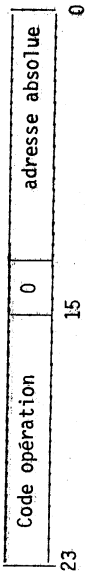
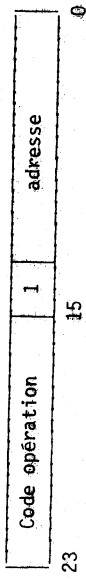


Fig.III.2:Format général d'une instruction

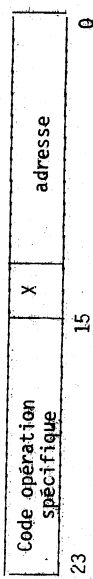
* adressage absolu :



* adressage indirect :



* adressage indexé :



X = 0 : registre d'index = D
X = 1 : registre d'index = E

* adressage immédiat :

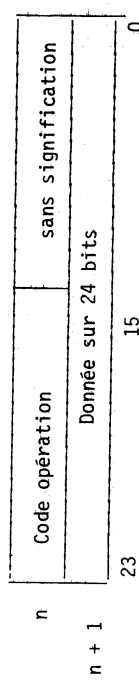


Fig. III.3 :

Modes d'adressage de GESPRO

La zone commande est formée d'une constante (le code opération) suivie, pour la plupart des instructions d'une zone complémentaire représentative des registres utilisés.

C. Déroulement d'une instruction

L'exécution d'une instruction peut se résumer de la manière suivante :

- * examiner s'il y a une requête d'interruption
- si oui : passer au changement de contexte pour prendre en compte l'interruption
- si non : passer en phase de recherche de l'instruction, ce qui implique une lecture mémoire à l'adresse indiquée par le compteur d'instructions
- * récupérer l'instruction et la décoder, ce qui permet de définir le schéma d'adressage associé (direct, indirect, indexé, immédiat, plusieurs mots)
- * dérouler une séquence de microinstructions déterminée correspondant à l'exécution proprement dite.

a) Cas particulier de l'architecture utilisée

La phase d'examen de requête d'interruption est effectuée par passage en ligne 0 colonne 15 de la mémoire de micro-instructions (cf. description du MCU).

L'unicité de la case mémoire autorisant l'interruption implique que toutes les instructions ont une séquence commune de recherche.

Celle-ci consiste à adresser la mémoire centrale. Un décodage partiel est effectué au niveau du MCU par l'utilisation d'un saut conditionnel JPX opérant sur les 4 bits les plus significatifs du code opération. Ceci provoque un débanchement vers 16 possibilités différentes, ce qui signifie que nous disposons de 16 groupes principaux d'instructions.

D'autres structures de saut permettent ensuite un nouveau débanchement en fonction des 4 bits les moins significatifs du code opération (JPR, JRL, JLL).

Les figures III.4 et III.5 illustrent ce que nous venons d'expliquer.

b) Définition du jeu d'instructions

Elle passe par la définition de la séquence (commune) de recherche d'instruction et de la procédure de changement de contexte en cas d'interruption. De plus on définit les possibilités d'adressage (voir fig. III.3)

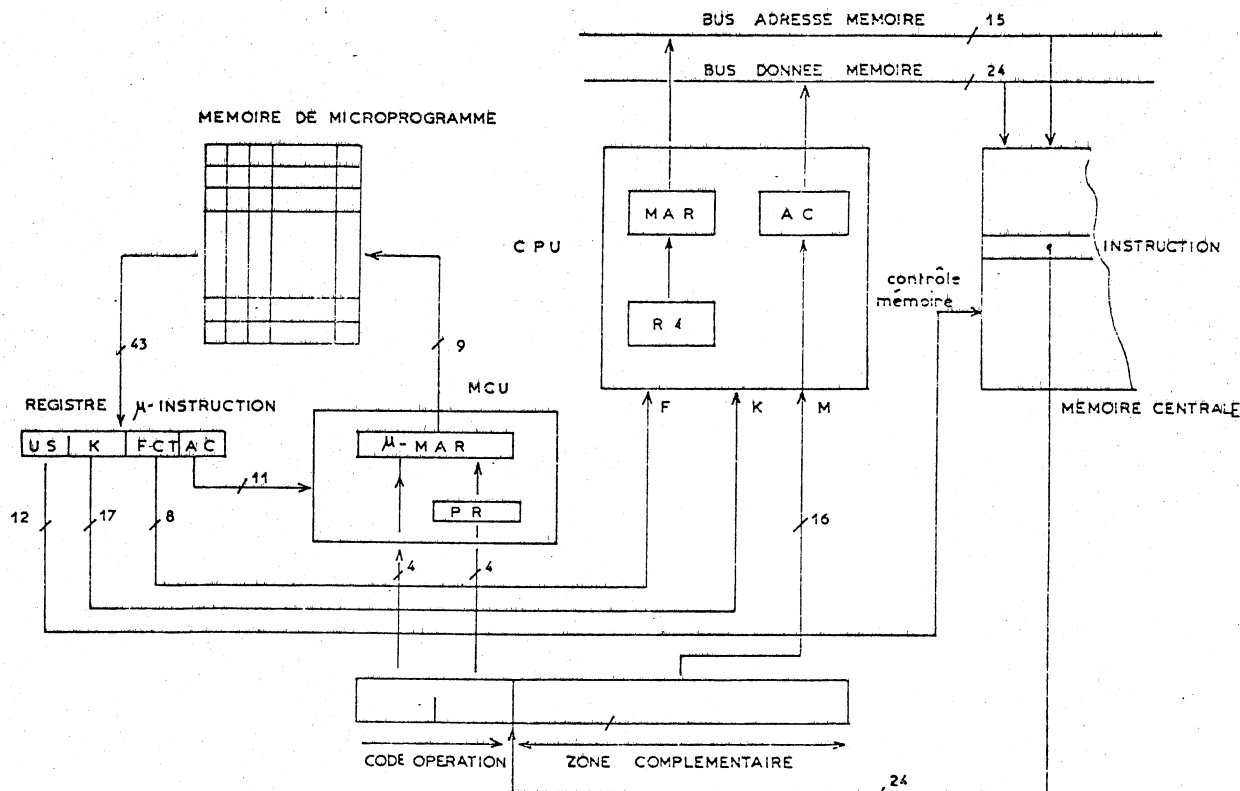


Fig. III.4. : SYNOPSIS DU DÉROULEMENT D'UNE INSTRUCTION.

- direct (puisque la zone complémentaire le permet)
- indirect (il existe un modificateur câble)
- indexé, ce qui nécessite la définition des registres concernés

Les registres concernés sont les suivants :

- deux accumulateurs : A et B (R_0 et R_1)
- deux registres d'index : D et E (R_2 et R_3)

Il est également nécessaire de disposer d'un registre compteur de programme : R_4 .

Avec cette structure de base il est possible de définir le jeu d'instructions que l'on veut réaliser.

c) Remarque : Rôle des modificateurs d'instructions

Nous avons vu que les instructions avaient accès à 4 registres : R_0 , R_1 , R_2 et R_3 . Avec le codage des micro-opérations de la machine INTEL 3000 il faudrait écrire autant de microprogrammes d'interprétation d'une opération que le nombre de registres qui peuvent être utilisés dans cette opération.

Pour pallier à cet inconvénient, on a adjoint deux bits opératoires appelés modificateurs d'instructions dont nous allons examiner le rôle.

Nous avons vu que la partie basse du code opération peut contenir un ou deux numéros de registres utilisés dans l'opération. Les bits F_0 à F_3 étant le codage du numéro du registre concerné par la micro-opération, le microprogramme écrira les microinstructions d'interprétation d'une instruction pour le registre R_0 seulement ; lorsque dans la microinstruction les modificateurs d'instructions MM ou MP seront positionnés à 1 les fonctions F_0 à F_3 seront masquées avec la partie basse du code opération.

Exemple : Les instructions LD, RN, ADR avec $n = 0, 1, 2, 3$ qui correspondent au changement du contenu de la case ADR dans le registre R_n diffèrent par le code opération qui est donné par $320 + R_n$ mais correspondent à un même microprogramme d'interprétation.

D. Outils de génération d'instructions

Dans un premier temps on écrit les micro-programmes de manière simplifiée en :

- faisant la liste des micro-opérations
- définissant le contenu du champ de contrôle des indicateurs

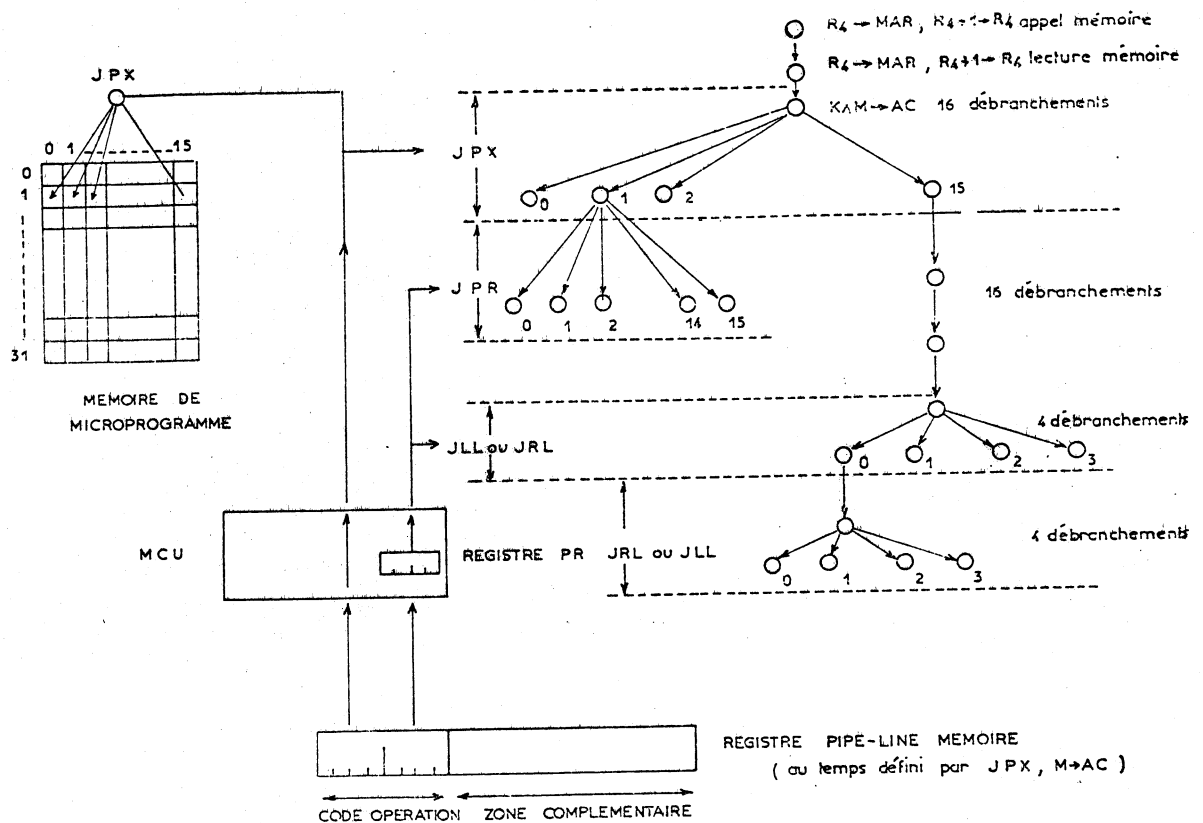


Fig. III.5: SYNOPSIS DU DECODAGE DE L'INSTRUCTION.

- définissant les opérateurs câblés mis en oeuvre (mémoire, module "multiplieur", module d'interruptions, etc...) et prévoyant les microcommandes associées

Une seconde phase consiste à définir des groupes d'instructions suivant des critères tels que :

- séquence(s) commune(s) pour l'ensemble des instructions du groupe
- mode d'adressage utilisé, car celui-ci est lié partiellement à un opérateur câblé (adressage indexé, indirect)

A ce stade de la conception du jeu d'instructions, nous avons suffisamment d'informations pour aborder la microprogrammation effective des instructions.

La microprogrammation consiste essentiellement à définir :

- le schéma d'implantation
- une image de la mémoire de microinstructions
- la structure des sauts permettant de générer les séquences en fonction de ce qui a été réalisé auparavant (instructions déjà écrites et implantées)

Le test des nouvelles instructions est fait en trois temps généralement :

- implantation de la séquence dans la mémoire de microinstructions vierge (on évite d'abord les problèmes de coexistence et d'implantation d'une nouvelle instruction avec un environnement existant) et test de l'algorithme utilisé
- vérification de la structure des sauts, ce qui consiste à voir si la suite des micro-instructions permet toujours un retour en phase de recherche de l'instruction
- vérification du fonctionnement de l'instruction en la plaçant dans un contexte connu, où le résultat de l'instruction est prévu.

SINTRAN III : système d'exploitation du mini-ordinateur NORD 10, disposant de facilités d'édition (QED) de programmation (BASIC, FORTRAN) et de gestion de fichiers, a été largement utilisé.

Nous utilisons un fichier disque (UMEM 33) contenant une image de la mémoire de micro-instructions correspondant aux instructions ou séquences déjà testées (fichier utilisé comme version définitive) et des fichiers de travail correspondant à des instructions (ou groupes d'instructions) en cours d'écriture ou de test ou à une nouvelle mouture du set existant en vue de faciliter l'implantation de nouvelles instructions.

Le format des enregistrements du fichier est le suivant :

- l'adresse physique (décimale) en mémoire de micro-instructions, suivie de 4 mots (en octal) obtenus par découpage en zones de 12 bits de la micro-instruction

Une représentation matricielle (32 lignes, 16 colonnes) purement formelle de la mémoire de microinstructions permet de vérifier le contrôle des adresses des microinstructions d'une séquence donnée.

L'implantation d'instructions nouvelles nécessite la connaissance de la topographie de la mémoire de microprogrammes. Un programme BASIC appelé MAP permet de générer un tableau d'occupation de cette mémoire correspondant à n'importe quel fichier (UMEM 33 ou autre). Les tests du jeu d'instructions sont faits en interactif avec le mini-ordinateur NORD-10. Un programme BASIC appelé UMRW permet d'effectuer les opérations nécessaires suivantes sous forme de fonctions :

- STOP : arrêter GESPRO
- UMW : écrire en mémoire de microinstructions
- UMR : lire en mémoire de microinstructions
- MM : écrire en mémoire centrale
- MR : lire en mémoire centrale
- LTES : charger en mémoire de microinstructions un fichier quelconque (UMEM 33 ou autre)
- LAM : lire un bloc de la mémoire centrale entre une adresse initiale et une adresse finale
- START : démarrer GESPRO

Ces fonctions sont toutes exécutées via CAMAC.

E. Le jeu d'instructions^{11,12)}

On peut distinguer d'une part un jeu d'instructions classiques dans lequel se trouvent les instructions rencontrées dans la majorité des jeux existants et enrichi d'instructions faisant utilisation de l'architecture spécialisée de GESPRO (Entrées-Sorties, Interruptions), d'autre part un ensemble d'instructions spécialisées destinées à accroître les performances et faciliter l'utilisation.

e) le traitement des interruptions

Un ensemble d'instructions est spécialisé au traitement des interruptions. Il permet les tâches suivantes :

- activer ou désactiver le module interruption
- masquer ou démasquer un niveau
- armer un niveau
- lire le niveau courant

f) les instructions "système"

Pour faciliter la tâche de l'utilisateur, un certain nombre d'instructions a été étudié. Elles concernent :

1) l'initialisation de GESPRO

Cette procédure a pour effets :

- de remettre à zéro tous les registres extérieurs à l'Unité Centrale (registre donnée-mémoire....)
- de désactiver les opérateurs câblés ainsi que le système de gestion des interruptions

Pour ce dernier, toutes les interruptions sont effacées, tous les

niveaux sont masqués à l'exception du niveau le moins prioritaire (niveau zéro), actif en permanence

2) le changement de contexte

L'abandon d'une tâche au profit d'une autre plus prioritaire

implique :

- la prise en compte de l'interruption associée
- la sauvegarde de l'état machine actuel (les deux accumulateurs A et B, les deux registres d'index D et E, "C" indicateur de dépassement, le compteur d'instructions R_4).

Les opérations de sauvegarde et de chargement en mémoire de l'état machine sont appelées "changement de contexte".

g) les instructions spécialisées

Ces instructions sont spécialisées dans le traitement des opérateurs câblés adjoints à GESPRO (multiplications, conversions, addition flottante)

Remarque : Le jeu d'instructions dans son état actuel est donné en annexe A.

II. LE SYSTEME DE DEVELOPPEMENT DE PROGRAMMES ¹³⁾

A. Introduction

Un système d'exécution de programmes a été élaboré pour GESPRO, système dont le support principal est le mini-ordinateur NORD-10, permettant l'utilisation de l'édition de texte et de la mémoire de masse principalement. Il assume les tâches suivantes :

- la gestion des jeux de définition d'instructions
- l'édition de texte
- l'assemblage à partir de jeux d'instructions adaptés à l'expérience
- l'édition de liens entre modules relogeables
- le chargement par CAMAC dans la mémoire centrale de GESPRO des modules relogés
- le contrôle mémoire de GESPRO
- une option "debugging", avec possibilité d'exécution à partir de NORD-10.

B. Généralités

Le langage de programmation le plus élémentaire, c'est à dire le plus proche du langage machine (binaire) est le langage assembleur et un programme qui traduit celui-ci en langage machine s'appelle aussi assembleur. Un assembleur traduit un module source (en langage symbolique) en un module objet (en langage machine) instruction après instruction, en un ou plusieurs passages.

L'objet produit est dit binaire absolu s'il est chargeable en mémoire centrale (en vue d'exécution) à une adresse fixe. Mais un tel objet est rigide, c'est pourquoi la quasi-totalité des assembleurs produit des programmes dits en "format binaire relogeable" (B.R.F.), c'est à dire chargeables en mémoire centrale à une adresse quelconque, après une édition de liens entre les différents modules constitutifs du programme.

L'éditeur de liens, à partir d'une adresse de chargement, calcule et translate les adresses à l'intérieur de chaque module de façon à relier et à placer les unes derrière les autres les différentes parties créant ainsi un module exécutable (voir fig. N° III.6).

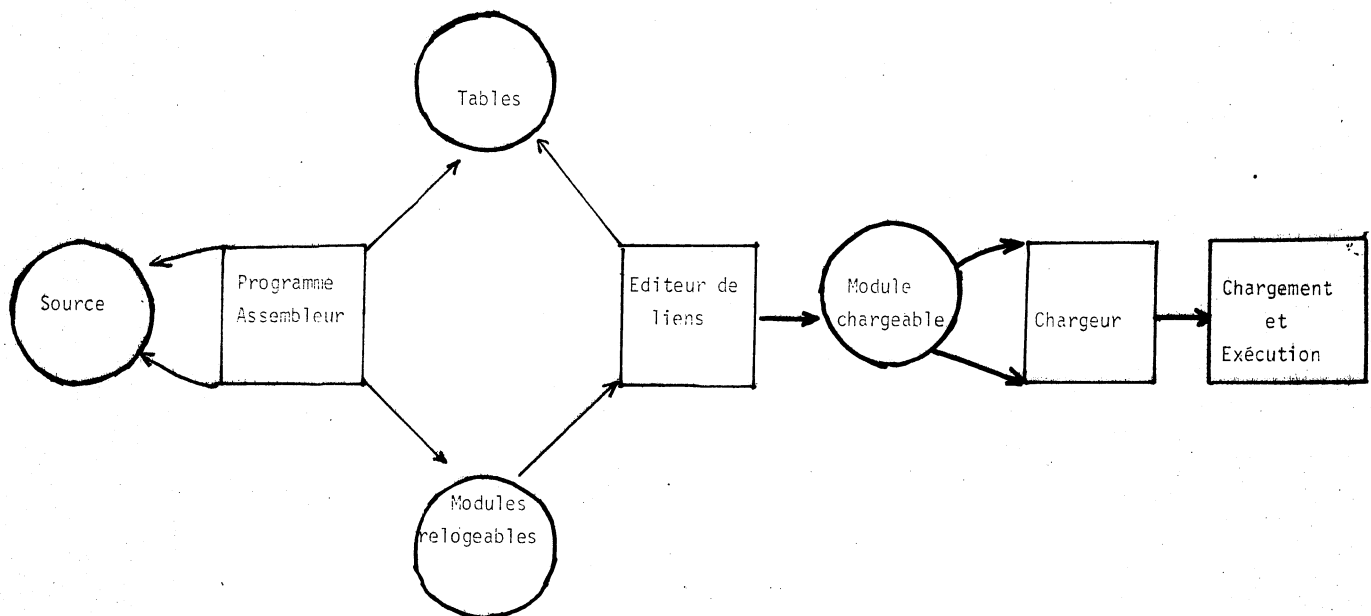


Fig.III.6

Fonctions de l'assembleur, éditeur de liens et chargeur

L'élaboration d'un programme de l'écriture à l'exécution passe par trois stades fondamentaux au cours desquels il est stocké en mémoire de masse

- avant assemblage, au niveau du langage symbolique, les programmes sont stockés sous forme de module source. L'avantage de ce stockage est que l'on dispose de programmes facilement modifiables et intelligibles, mais par contre non directement chargeables
- après assemblage, au niveau du binaire relogeable, les programmes sont stockés sous forme de module objet. A ce stade les programmes ne sont ni modifiables aisément ni chargeables directement
- après l'édition de liens, au niveau du binaire absolu, les programmes sont chargeables directement à l'adresse fixée lors de l'édition de liens ; ils sont stockés sous forme de modules images-mémoire ou modules chargeables. A ce niveau les programmes ne sont pas modifiables facilement (ce travail est cependant faisable grâce à l'option "debugging" du chargeur). Ce type de stockage est destiné à des programmes testés et exécutés toujours à la même adresse en mémoire centrale de GESPRO

En résumé :

- les programmes non testés et modifiables sont stockés sous forme de modules sources
- les programmes testés mais chargeables à une adresse variable sont stockés sous forme de modules objets
- les programmes testés et chargeables à une adresse fixe sont stockés sous forme de modules chargeables.

Les différents éléments logiciels du système de développement au niveau opérationnel se répartissent comme l'indique la figure III.7.

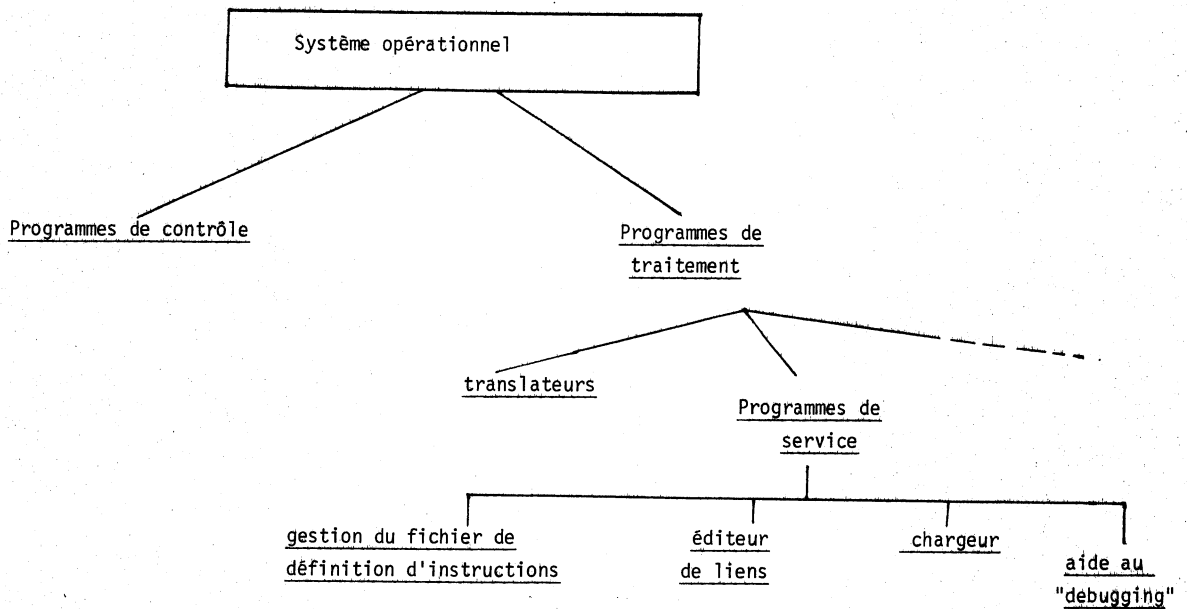


Fig. III.7 :

Structure logicielle d'un système de développement dans le cadre du système d'exécution d'un ordinateur

C. Structure du système

a) Gestion du fichier de définition d'instructions

1) Introduction

Au fur et à mesure de l'avancement de l'expérience, les algorithmes de sélection des événements vont évoluer, changer, se raffiner, ce qui va se traduire par l'écriture et l'implantation de nouvelles instructions spécialisées : le jeu d'instruction de GESPRO sera en constante évolution, ce qui implique une gestion dynamique du fichier de définition d'instructions, fichier pris en compte par le programme d'assemblage

2) Caractéristiques générales

L'assembleur produit un programme en format binaire en fonction de représentation symbolique d'adresses et de codes opérations. A partir du mnémonique de l'instruction, il reconnaît la syntaxe et en partie la sémantique correspondante. Pour ce faire il dispose du jeu d'instructions sous une forme codée et compréhensible par lui, donnant la définition de chaque instruction. C'est ce qu'on appelle le fichier de définition d'instructions. Les mnémoniques sont rangés par ordre alphabétique par commodité pour l'utilisateur et pour le programme de gestion.

La structure du programme de gestion du fichier de définition d'instructions est analogue à celle de Q.E.D (éditeur de texte de NORISK DATA) : ainsi il possède les caractéristiques suivantes :

* manipulations générales sur un fichier :

- création physique
- effacement physique
- impression du contenu sur un terminal quelconque
- copie d'un fichier dans un autre

* manipulations générales sur le contenu d'un fichier :

- adjonction et effacement de ligne
- impression de ligne
- modification de ligne
- traitement de caractères et d'expressions

b) Assembleur

En fait c'est un cross-assembleur dont le support est le mini-

ordinateur NORD-10 avec son système opérationnel SINTRAN III. La structure suivante a été adoptée en vue d'obtenir un assemblage performant du point de vue de la vitesse d'exécution principalement

- assemblage en un seul passage
- écriture du programme en langage assembleur NORD exécuté sous SINTRAN III
- utilisation de fichiers NORD de façon à profiter au maximum des facilités de gestion de fichiers de SINTRAN III (aussi bien pour les fichiers sources que pour les fichiers objets et temporaires).

La particularité saillante de cet assembleur est sa capacité à pouvoir travailler sur des jeux d'instructions différents.

Le programme recherche les erreurs :

* au niveau de l'instruction :

- existence d'un code opération ou d'une directive
 - syntaxe de l'instruction
- * au niveau d'un programme :
- existence des directives de début et de fin de programme
 - indéfinition de symboles

* au niveau de l'assembleur :

- dépassement de capacité des tables
- "overflow" des constantes et des adresses

Cet assembleur offre une large gamme d'utilisations :

- assemblage en interactif
- assemblage avec ou sans listing
- assemblage avec listing partiel
- assemblage conditionnel
- listing ou non des tables
- listing des variables
- option de "debugging" grâce à un jeu de diagnostics

c) Chargeur

Le chargeur représente la phase finale du système de développement de programmes. A partir des modules objets créés lors de l'assemblage, il

permet de constituer physiquement un programme exécutable en mémoire. Il est composé des fonctions principales rencontrées dans tout chargeur relogeable :

- * relier les différents modules constituant un programme pour en faire un module relogeable
- * reloger le programme en fonction de l'adresse de chargement
- * placer physiquement le programme dans la mémoire centrale de GESPRO

A ces fonctions s'ajoutent des fonctions spécifiques à GESPRO :

- * contrôle du statut de GESPRO avant chaque chargement
- * système de recherche d'erreur élaboré
- * contrôle et visualisation de l'implantation en mémoire centrale de GESPRO
- * option de debugging sur les programmes chargés

III. LE SYSTEME D'AIDE A LA MICROPROGRAMMATION

A. Introduction

Les principes de ce logiciel découlent des considérations suivantes :

- le micro-programmeur doit pouvoir écrire ses microprogrammes en se consacrant exclusivement aux algorithmes (éviter d'écrire des programmes sous forme binaire). Pour résoudre ce problème, il faut donner au micro-programmeur un langage symbolique lui permettant de coder les sections et l'enchaînement des micro-instructions de façon claire et synthétique
- le micro-programmeur n'a pas à être concerné par l'implantation des micro-instructions en mémoire de commande, surtout lorsque la logique de séquençement augmente la complexité de cette tâche.
- les algorithmes des microprogrammes doivent pouvoir être mis au point facilement. Pour cela il est nécessaire de donner au microprogrammeur un simulateur fonctionnel du déroulement de l'exécution des micro-programmes.

On veut de plus optimiser l'occupation de la mémoire de commande en utilisant les possibilités offertes par la logique de séquençement.

B. Choix du langage de microprogrammation

Les critères de choix entre un langage de microprogrammation du type assembleur et un langage plus "évolué" ont été guidés par la comparaison entre programmation et microprogrammation : la microprogrammation se distingue de la programmation par le parallélisme dans les actions, la complexité des branchements, l'exigence d'efficacité et la proximité de la structure de la machine. Un langage de programmation évolué offre des structures de contrôle et des structures de données utiles pour transcrire des algorithmes structurés mais mal adaptées pour décrire le parallélisme et les branchements complexes des algorithmes microprogrammés. Ces raisons ont guidé le choix d'un langage du type assembleur qui sera aussi le langage source du simulateur fonctionnel.

C. Présentation du système

Le système permet les opérations suivantes :

- 1) la simulation par programme (avec comme entrées les codes opérations des instructions existantes et les microprogrammes associés) de l'exécution de tous les microprogrammes
- 2) la traduction en binaire des microinstructions données dans un langage source du type assembleur
- 3) l'implantation en mémoire de commande des microprogrammes d'interprétation des instructions GESPRO
- 4) l'optimisation de la place occupée en mémoire c'est à dire la réutilisation des microinstructions existant déjà en mémoire et ayant un contenu semblable à celui des micro-instructions à implanter.

Cette dernière fonction n'est pas utopique : pour le jeu d'instructions existant une étude a permis de savoir que le taux de réutilisation de chaque microinstruction était de trois. Une économie du nombre de plans de mémoire de commande est importante pour la vitesse de l'ensemble du système qui dépend du temps d'accès à la mémoire de commande (ce temps augmente en même temps que la taille de la mémoire et donc de celle de l'adresse mémoire : à cause du décodage d'adresse demandant des couches technologiques supplémentaires). Le logiciel d'implantation choisit le code opération qui conduit à la "meilleure" réutilisation. Son choix est guidé par les renseignements suivants :

* Le rôle de la partie basse du code opération ; ce rôle est fourni par le microprogramme et peut être :

- désigner deux registres
- désigner un registre et distinguer l'instruction dans le groupe auquel elle appartient
- caractériser l'instruction

* la liste des codes opérations déjà utilisés

* le classement par le microprogramme des instructions en groupes fonctions des algorithmes d'interprétation et l'indication du ou des groupes éventuels auxquels l'instruction est susceptible d'appartenir

Le système d'implantation tient à jour :

- la liste des codes opérations déjà utilisés
- la liste des groupes d'instructions avec les codes opérations des instructions de chaque groupe
- l'étiquetage de l'ensemble des micro-instructions avec les numéros des groupes utilisant chaque microinstruction

Le système est interactif pour permettre au microprogrammeur de résoudre certains choix en lui laissant la possibilité d'implanter lui-même ses microprogrammes en mémoire.

CHAPITRE IV

TEST DE L'APPAREILLAGE

I. INTRODUCTION

Les tests concernant les modules suivants : les opérateurs câblés adjoints à GESPRO et la mémoire GESPRO :

* le module "multiplicateur" avec ses sous-modules :

- conversion entier-réel
- conversion réel-entier
- ajustage de réels
- multiplication d'entiers
- multiplication de réels

* le module de recherche du plus grand moment transverse

* la mémoire

II. TEST DES OPERATEURS CABLES PAR NORD-10

Dans un premier temps, les modules ou sous-modules sont testés par NORD-10 seul, via CAMAC. La configuration de test est donnée sur la figure IV.1. Les programmes sont écrits en langage assembleur (tout au moins pour ce qui concerne la partie entrées-sorties CAMAC et le lancement de l'opération spécifique à un module) pour gagner en temps d'exécution des programmes et pour soumettre les modules à un rythme élevé d'opérations.

L'opération effectuée par un module est simulée dans le même programme par NORD 10 ; la confrontation des deux résultats permet de diagnostiquer les erreurs.

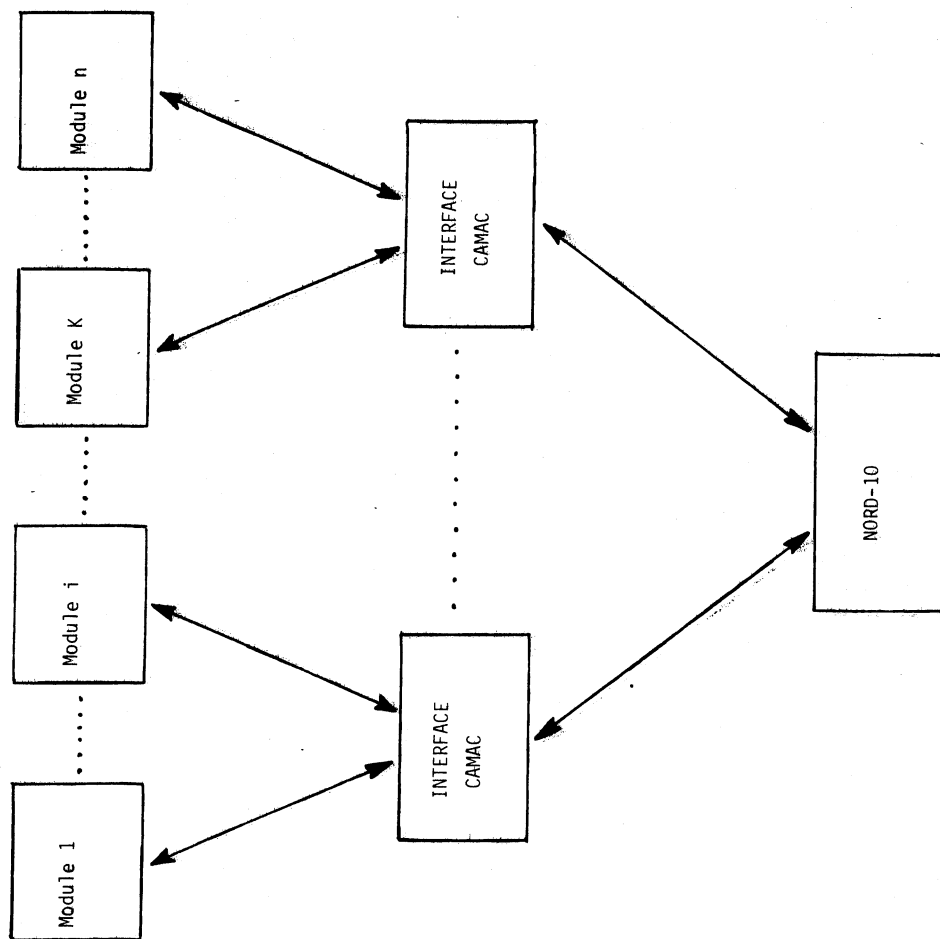


Figure IV.1 : Configuration de test impliquant NORD-10 seul

III. TEST DES OPERATEURS CABLES PAR GESPRO

Les tests les plus sévères sont effectués par GESPRO : ils sont microprogrammés, ce qui fait que les modules sont sollicités à la fréquence maximale, égale ou même supérieure à celle demandée dans la configuration expérimentale.

Le résultat d'une opération microprogrammée est sauvegardé dans la mémoire centrale de GESPRO, dans une zone dite "boîte aux lettres", où il est récupéré par NORD-10, en vue de le comparer avec le résultat calculé par lui. Une case de la boîte aux lettres est réservée pour la synchronisation des tâches des deux machines : suivant la valeur de cette case "flag" GESPRO attend ou non que la tâche NORD-10 soit finie.

La configuration du test est donnée par la figure IV.2.

IV. LES TESTS MEMOIRE

19,20)

Il est nécessaire de tester sévèrement la mémoire :

- parce que le constructeur ne fait qu'un test statistique sur ses produits : on n'est donc jamais assuré du fonctionnement d'un composant (ceci est vrai pour tous les circuits intégrés, pas seulement les mémoires)
- parce que les opérations à référence mémoire forment la grande majorité des opérations effectuées par un ordinateur.

A. Généralités sur les mémoires

Le diagnostic d'un élément mémoire requiert l'exécution de longues et complexes séquences de test, séquences concernant les listes d'adresses, les états logiques de commande appliqués à un ensemble de cellules élémentaires d'une mémoire. Le contenu d'une mémoire dépend de ses contrôles externes qui, seuls normalement, doivent modifier le contenu d'une ou de plusieurs cellules. Le contenu peut être permanent ou altérable ; quand il est permanent, le composant est appelé à lecture seule ou "read-only memory" (ROM) quand il peut être modifié, il est appelé mémoire à lecture-écriture ou "read-write memory".

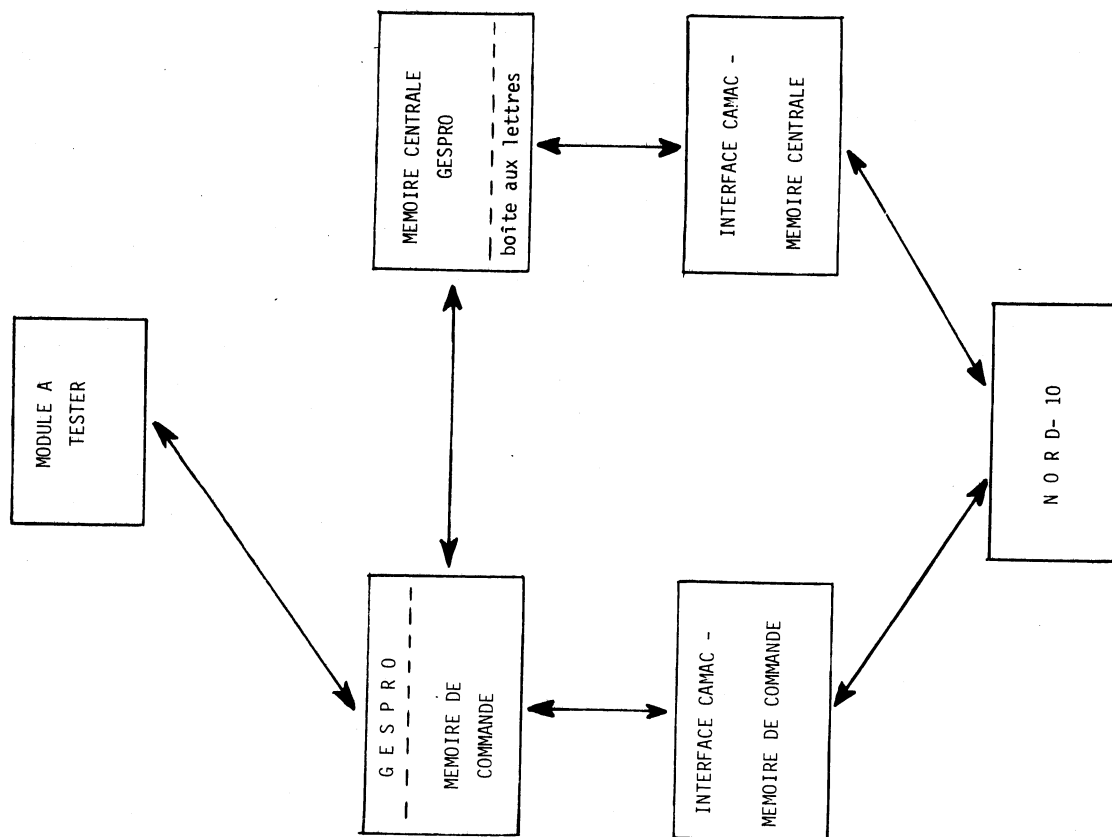


Figure IV.2 : Configuration de test impliquant NORD-10 et GESPRO

Le mode d'accès de chaque cellule d'une mémoire la caractérise de la façon suivante : si une cellule n'est adressée qu'après celles qui la suivent ou la précèdent elle est appelée "mémoire-série" ou "mémoire à accès séquentiel", par exemple, le registre à décalage ; si une cellule peut être adressée indépendamment de la précédente adressée, elle est appelée à "mémoire à accès aléatoire", par exemple, la RAM (random access memory), qui peut être à lecture seule ou à lecture-écriture.

En général on réfère comme RAM une mémoire à écriture-lecture et à accès aléatoire, réservant le nom de ROM aux mémoires à lecture seule, alors qu'une ROM peut être à accès séquentiel ou à accès aléatoire.

Une des caractéristiques importantes d'une mémoire est sa faculté d'avoir ou de ne pas avoir son contenu modifié par sa lecture. Par exemple, une mémoire à lecture non destructive permet une lecture répétitive sans que son contenu ne soit modifié. Par comparaison une mémoire à tores doit être réécrite après sa lecture. Certains types de mémoires perdent leur contenu en fonction du temps écoulé après leur écriture. Ces mémoires peuvent donc être aussi classées en fonction du temps de rétention de leur contenu. Certains types de mémoires statiques conservent leurs informations tant que l'alimentation n'est pas coupée. D'autres retiennent leurs informations sans nécessiter un cycle de réécriture (refresh), et même lorsque leurs sources d'alimentation sont coupées.

Certains mémoires dynamiques conservent leurs informations dans leurs capacités propres. Elles ne les conservent que durant quelques millisecondes, elles doivent donc être "rafraîchies" (éventuellement par un cycle de lecture) si leur contenu doit être conservé longtemps.

D'où la nécessité de tester les conditions dans lesquelles la mémoire se comportera d'une façon indésirable : caractéristiques électriques non conformes, réactions intempestives, etc...

Les défauts rencontrés peuvent être groupés en quatre catégories :

- * Absence d'une ou plusieurs cellules
- * Perturbation d'une cellule par certaines de ses voisines
- * Dépendance du temps d'accès d'une cellule avec l'adresse
- * Ambiguïté d'adresse d'une cellule.

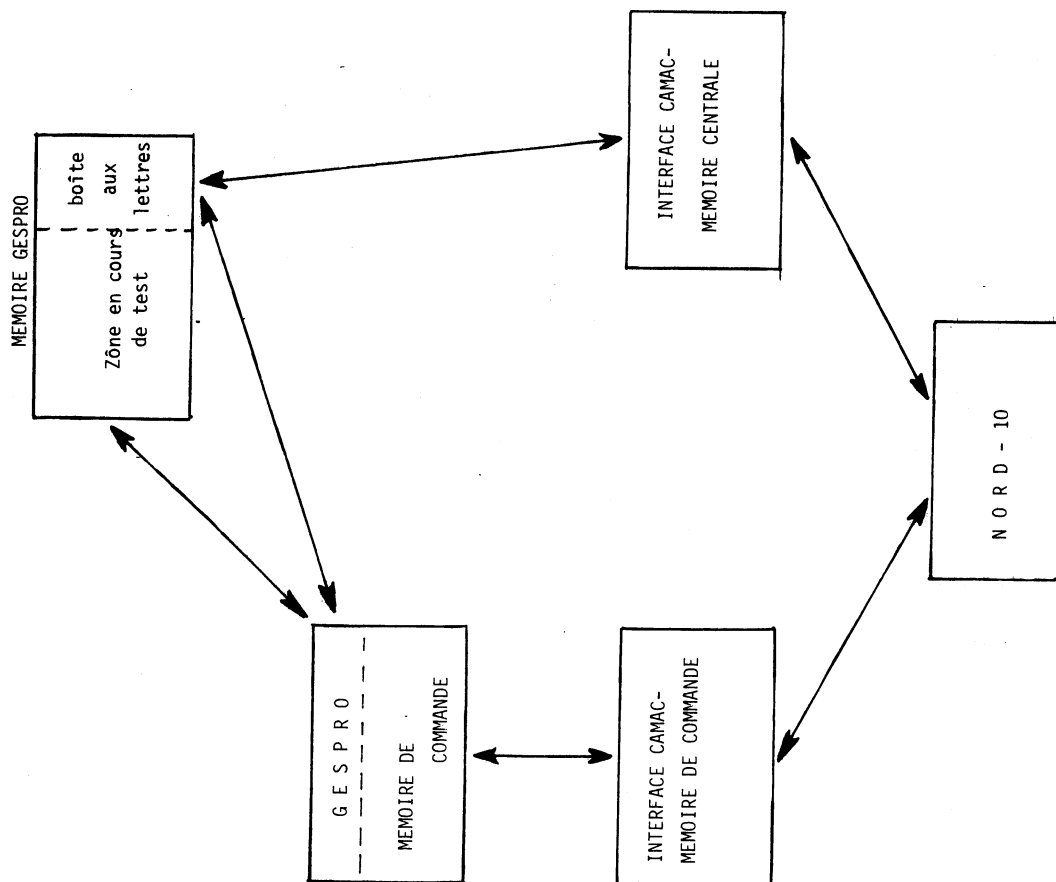


Figure IV.3 : Configuration générale d'un test mémoire

Nous allons décrire une série de tests permettant, en principe, de mettre en évidence ces défauts.

La meilleure solution est, connaissant la topographie de la mémoire d'en tester les points sensibles, dus à des couplages inter-cellules ou à l'influence des décodeurs et des drivers sur les cellules avoisinantes. On peut ainsi réduire les séquences de test à celles qui ont le plus de chance de mettre des défauts en évidence. Toutefois, si cette connaissance existe au niveau du fabricant, elle est rarement fournie à l'utilisateur.

Ne connaissant pas la topographie de la mémoire (les cellules ne sont pas forcément organisées géographiquement dans le même ordre que l'ordre numérique), nous avons développé des programmes généraux.

La mémoire que nous avons à tester est une RAM statique de 8 K mots de 24 bits. La configuration de test est donnée sur la figure IV.4. Les micro-programmes exécutés par GESPRO arrêtent le test en cas d'erreur et sauvegardent l'état des cases de la mémoire qui constituent l'erreur dans la boîte aux lettres (partie de la mémoire réservée au dialogue GESPRO-NORD 10) où il est récupéré par un programme NORD-10 qui relance le test aussitôt après.

B. Défauts testés

a) Absence de cellule(s)

Pour détecter une cellule qui ne peut être ni écrite, ni lue, par les circuits extérieurs à la mémoire, nous utilisons un test appelé "WALKING ONE" déplaçant un UN à travers toute la mémoire, toutes les autres cellules contenant un ZERO.

Ce test comporte les étapes suivantes :

- i) écrire toutes les cellules avec des ZERO
- ii) écrire la première cellule avec un UN
- iii) lire toutes les autres cellules et vérifier qu'elles contiennent ZERO
- iv) vérifier que la cellule A contient toujours UN
- v) déplacer le UN dans la deuxième cellule
- vi) écrire ZERO dans la première cellule
- vii) reprendre la procédure à l'étape (iii)

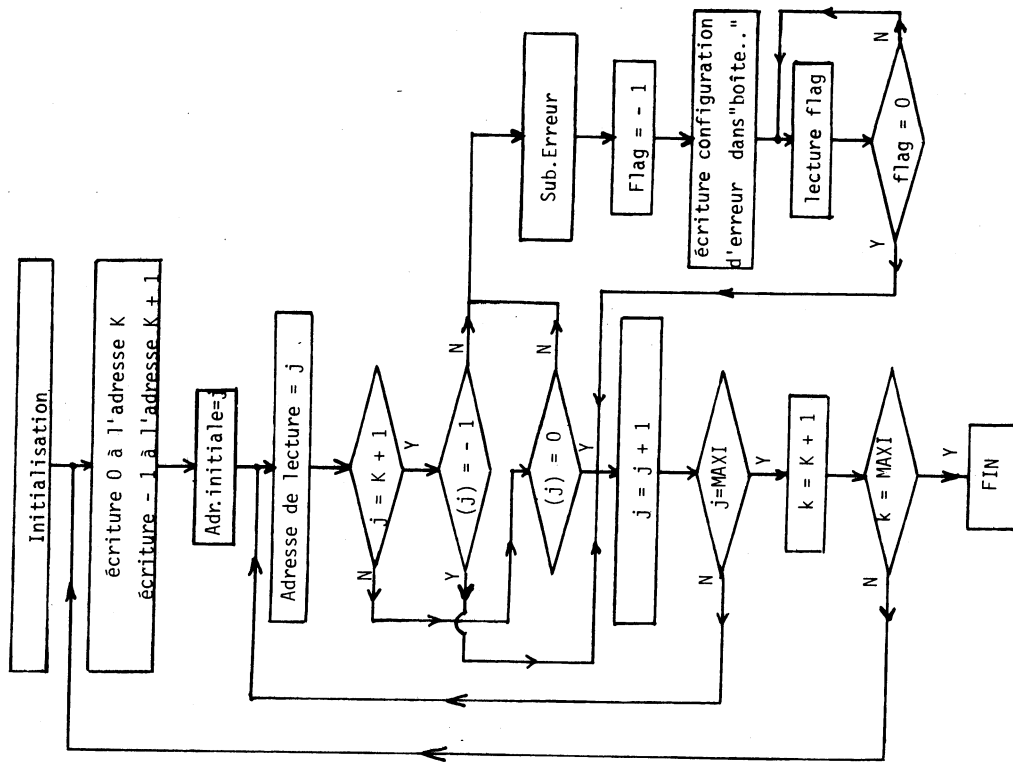


Figure IV.4 : Organigramme du test "WALKING-ONE"

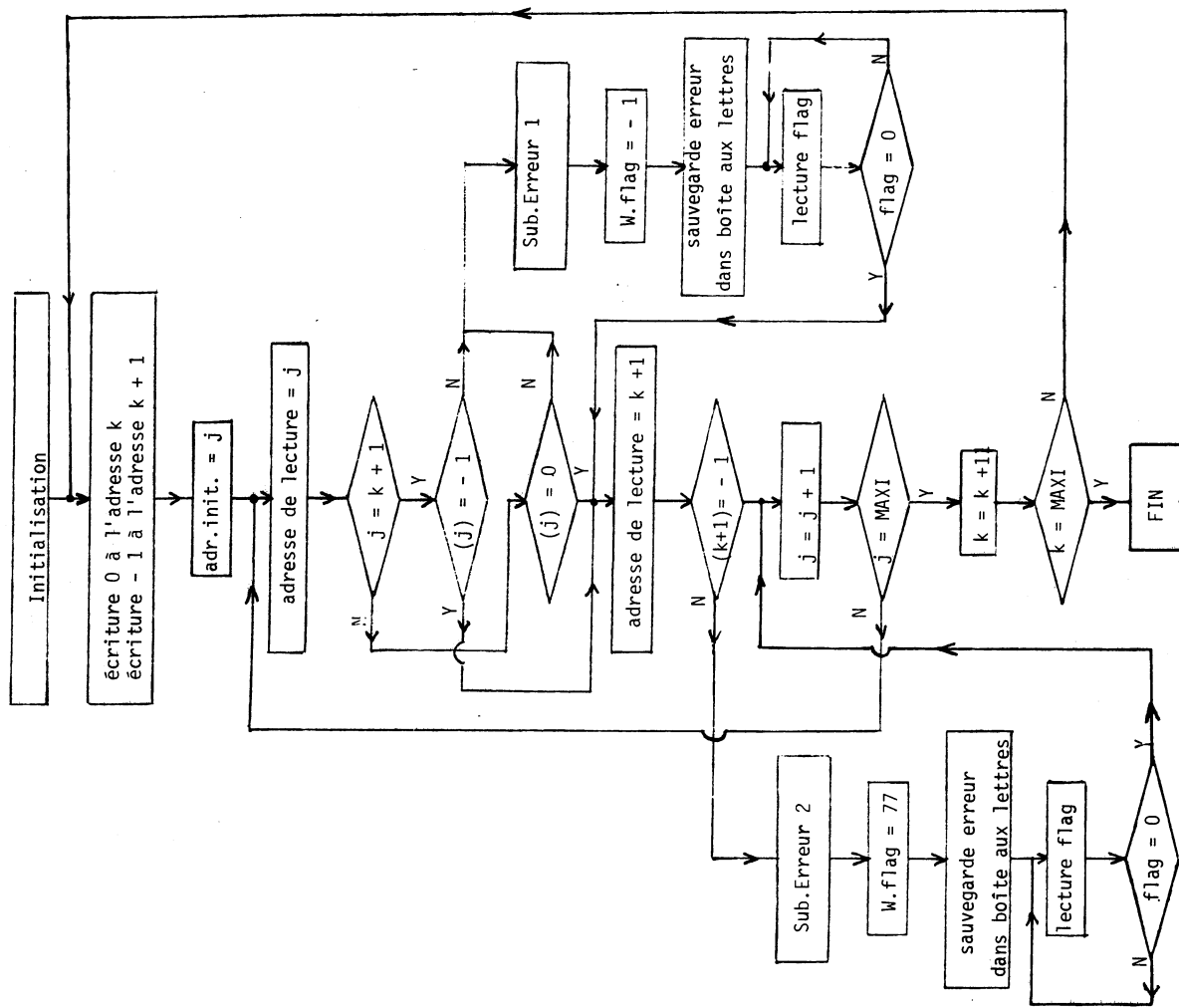


Figure IV.5 : Organigramme du test : "GALOPPING-ONE"

De la même façon nous avons utilisé un "WALKING ZERO".

La figure IV.5 donne l'organigramme d'un test du type "WALKING ONE".

Le nombre de cellules à tester étant N, le nombre de tests écritures et lectures est de $N^2 + 2N$.

b) Perturbation d'une cellule par l'état de ses voisines

Il s'agit ici de détecter le cas où une cellule est influencée par le contenu ou la variation du contenu de ses voisines. Le test consiste à entourer chaque cellule adressable par des cellules complémentées et de répéter l'opération. Pour ce faire on utilise la séquence dite "DISTURBED WALKING ONE". Elle consiste à :

- i) écrire UN dans une cellule A
- ii) écrire ZERO dans les autres cellules un certain nombre de fois
- iii) lire la cellule A, vérifier qu'elle contient UN
- iv) recommencer l'opération en (i) pour la cellule A + 1 (le cycle de perturbation et de contrôle est refait pour toutes les cellules de la mémoire).

Cette séquence (avec son complément "DISTURBED WALKING ZERO")

révélera les changements d'état d'une cellule, résultant d'un couplage parasite entre la cellule considérée et les cellules adjacentes.

c) Dépendance du temps d'accès d'une cellule avec l'adresse

La rapidité du changement des états internes des décodeurs d'adresse dépend de leur contenu antérieur. De ce fait on doit vérifier toutes les combinaisons d'adresses possibles en recherchant la combinaison qui produit le plus long temps d'accès de la mémoire.

Pour ce faire on utilise une séquence appelée "GALOPPING ONE" dont les étapes sont les suivantes :

- i) écrire toutes les cellules avec un ZERO
- ii) écrire la cellule A avec un UN
- iii) lire séquentiellement toutes les cellules (en vérifiant qu'elles contiennent ZERO) tout en revenant contrôler, pour chaque cellule vérifiée, que la cellule A contient toujours UN
- iv) écrire un ZERO dans la cellule A et un UN dans la cellule A + 1
- v) recommencer la procédure en (iii)

On déplace de cette façon le UN à travers toute la mémoire. De la même façon on utilise le test complémentaire "GALOPPING ZERO". Ainsi, chaque pas de la séquence donne accès à la cellule à vérifier à partir des emplacements de toutes les autres cellules.

La figure IV.6 illustre ce que nous avons expliqué. Le nombre de tests à effectuer est ici de $2N^2$ pour N cellules.

d) Ambiguïté d'adresse

Il faut détecter les cas où une même adresse décode plusieurs cellules et les cas où plusieurs adresses décodent la même cellule. Le test consiste à adresser séquentiellement chaque cellule et vérifier que d'autres cellules n'ont pas été écrites avec l'adresse de la cellule testée. Pour ce faire, on utilise la séquence "WALKING ONE" ou l'"ECHIQUIER" qui consiste à :

- i) écrire alternativement des UNS et des ZEROS dans toutes les cellules de la mémoire
- ii) lire complètement la mémoire et vérifier que les résultats sont conformes
- iii) refaire la même chose en échangeant les UNS et les ZEROS

C. Test NORD 10 et GESPRO

Les tests de la mémoire décrits plus haut permettent déjà de tester le double accès (GESPRO-NORD 10 en vol de cycle) de celle-ci. Pour tester plus sévèrement ce double accès nous avons, tournant en même temps :

- un microprogramme GESPRO ("WALKING-ONE", "GALOPPING-ONE" ou autre) travaillant sur une partie de la mémoire
- un programme NORD-10 écrit en langage assembleur travaillant sur une autre partie de la mémoire

Une dernière partie de la mémoire (la boîte aux lettres) permet la communication d'une machine à l'autre.

D. Performances des microprogrammes de test.

Nous avons mesuré les temps d'exécution de microprogrammes de test de la mémoire centrale ; nous avons obtenu, pour 8K mots de 24 bits

- pour WALKING ONE 1 mn 27 s
- pour GALOPPING ONE 2 mn 35 s

Ce type de test (les mêmes algorithmes) a été effectué sur un microprocesseur TMS 9900 de Texas, testant 8K mots de 16 bits de sa propre mémoire. Les performances obtenues sont les suivantes (les programmes sont écrits en langage assembleur TEXAS)

- 39 minutes pour WALKING ONE
- 54 minutes pour GALOPPING ONE

Ce facteur supérieur à 20 de différence entre les deux machines provient de la possibilité que nous avons utilisée de microprogrammer GESPRO, ainsi que du temps d'accès mémoire deux fois plus faible pour GESPRO que pour l'autre machine.

C O N C L U S I O N

Nous disposons d'un ensemble : GESPRO + opérateurs câblés rapides, capable de calculer en 20 microsecondes à peu près la masse d'un candidat dimuon et de prendre la décision de l'accepter ou de le rejeter. Cet ensemble très souple permet également de prendre une décision au vu de l'impulsion transverse des muons, de leur multiplicité (données disponibles grâce aux modules de recherche du plus grand moment transverse) et même le cas échéant des données complètes décrivant un événement (disponibles dans l'"EVENT BUFFER" : mémoire tampon vue par GESPRO comme un prolongement de sa propre mémoire).

Le dépouillement des données acquises actuellement sur l'expérience (avant l'installation de notre ensemble de filtrage) doit permettre de trouver les critères de filtrage les plus appropriés. La souplesse de votre ensemble permettra d'exécuter les algorithmes trouvés.

Un prototype testé est en état de fonctionnement ; les tests programmés et microprogrammés mis au point doivent permettre de faciliter la mise en oeuvre du dispositif complet. Le logiciel élaboré doit permettre la programmation d'algorithmes de filtrage appropriés à l'expérience.

A N N E X E JEU D'INSTRUCTIONS GESPRO 1) Remarques préliminaires

Le format des instructions est un ou plusieurs mots de 24 bits dont les 8 les plus significatifs du premier mot donnent le code opération. Les autres zones mémorisent soit des constantes soit des adresses (pour celles-ci : zone de 16 bits dont le plus significatif représente l'indexation).

Le programmeur dispose de 4 registres A, B, D et E dont 2 registres sont des accumulateurs et les deux autres des registres de base pour l'indexation.

La liste ci-dessous énumère les instructions actuellement implémentées avec les conventions suivantes :

- R_n représente le groupe des 4 registres généraux A, B, D et E ($0 \leq R_n \leq 3$)
- R_j représente le groupe des 2 accumulateurs A et B ($R_j = 0, 1$)
- R_i représente le groupe des registres de base D et E ($R_i = 0, 1$)
- ADDR est une adresse comprise entre 0 et 7777
- δ est le signe d'indirection
- DATA, CONF et MASK sont des données de 24 bits
- NBRE est un nombre compris entre 0 et 24 indiquant le nombre de décalages à effectuer
- l'adjonction des lettres S et D signifie respectivement "Source" et "Destination"
- ORDRE est l'adresse de l'ordre CAMAC spécifié sur 24 bits (IDC, F, B, C, N, A)

N.B. : Toutes les données de cette annexe sont en base 8

2) Le jeu d'instructions

1) Groupe : lecture/écriture mémoire

ST, R_n	ADR	104 + R_n	$(R_n) \rightarrow \text{ADR}$
ST, R_n	δ ADR	104 + R_n ; I = 1	$(R_n) \rightarrow (\text{ADR})$
STI, R_j, R_i	ADR	302 + R_j ; I = R_i	$(R_j) \rightarrow \text{ADR} + (R_i)$
LD, R_n	ADR	320 + R_n	$(\text{ADR}) \rightarrow R_n$
LD, R_n	δ ADR	320 + R_n ; I = 1	$((\text{ADR})) \rightarrow R_n$
LDI, R_j, R_i	ADR	300 + R_j ; I = R_i	$(\text{ADR} + (R_i)) \rightarrow R_n$

LDD, A	ADR	230	$(\text{ADR}) \rightarrow A$
LDD, A	δ ADR	230 ; I = 1	$(\text{ADR} + 1) \rightarrow B$
LDD, IE	ADR	231	$((\text{ADR})) \rightarrow A$
LDD, IE	δ ADR	231 ; I = 1	$((\text{ADR} + 1)) \rightarrow B$
MOVE	SADR, DADR, NBMOT	210	$(\text{ADR}) \rightarrow D$
STM	ADR, DATA	264	$(\text{ADR} + 1) \rightarrow E$
LIM, R_n	DATA	200 + R_n	$((\text{ADR})) \rightarrow D$
			$((\text{ADR} + 1)) \rightarrow E$
			$(\text{ADR} + i) \rightarrow \text{ADR} + i$
			i = 0, NBMOT
			DATA $\rightarrow \text{ADR} + i$
			i = 0, (A)
			DATA $\rightarrow R_n$

2) groupe arithmétique / logique à référence mémoire

a) groupe arithmétique

ADD, R_n	ADR	334 + R_n	$(R_n) + (\text{ADR}) \rightarrow R_n$
ADDI, R_j, R_i	ADR	314 + R_j ; I = R_i	$(R_j) + (\text{ADR} + (R_i)) \rightarrow R_j$
SUB, R_n	ADR	324 + R_n	$(R_n) - (\text{ADR}) \rightarrow R_n$
SUBI, R_j, R_i	ADR	316 + R_j ; I = R_i	$(R_j) - (\text{ADR} + (R_i)) \rightarrow R_j$
ADIM, R_j	DATA	212 + R_j	$(R_j) + \text{DATA} \rightarrow R_j$
SUBIM, R_j	DATA	207 + 8 * R_j	$(R_j) - \text{DATA} \rightarrow R_j$

b) groupe logique

AND, R_j	ADR	220 + R_j	$(R_j) \wedge (\text{ADR}) \rightarrow R_j$
AND, R_j	δ ADR	220 + R_j ; I = 1	$(R_j) \wedge ((\text{ADR})) \rightarrow R_j$
ANDI, R_j, R_i	ADR	306 + R_j ; I = R_i	$(R_j) \wedge (\text{ADR} + (R_i)) \rightarrow R_j$
ANDIM, R_j	DATA	204 + 8 * R_j	$(R_j) \wedge \text{DATA} \rightarrow R_j$

OR	R _j	ADR	222 + R _j	(R _j) V (ADR) → R _j
OR	R _j	⊙ ADR	222 + R _j ; I = 1	(R _j) V ((ADR)) → R _j
ORI	R _j , R _i	ADR	310 + R _j ; I = R _i	(R _j) V (ADR + (R _i)) → R _j
ORIM	R _j	DATA	205 + 8 * R _j	(R _j) V DATA → R _j
XOR	R _j	ADR	224 + R _j	(R _j) ⊕ (ADR) → R _j
XOR	R _j	⊙ ADR	224 + R _j ; I = 1	(R _j) ⊕ ((ADR)) → R _j
XORI	R _j , R _i	ADR	312 + R _j ; I = R _i	(R _j) ⊕ (ADR + (R _i)) → R _j
XORIM	R _j	DATA	206 + 8 * R _j	(R _j) ⊕ DATA → R _j
c) groupe : opération mémoire				
DCM	ADR		60	(ADR) - 1 → ADR saut à RAC + 1 si (ADR) ≠ 0
DCML, R _i	ADR		305; I = R _i	(ADR + (R _i)) - 1 → ADR + (R _i) saut à RAC + 1 si (ADR + (R _i)) ≠ 0
INM	ADR		62	(ADR) + 1 → ADR
INML, R _i	ADR		304; I = R _i	(ADR + (R _i)) + 1 → ADR + (R _i) saut à RAC + 1 si (ADR + (R _i)) ≠ 0
LBYT, R _n	ADR, MASK		330 + R _n	[(ADR) 1377 _g + (ADR) * 2 ⁸] ∧ MASK
LBYT, R _n	⊙ ADR, MASK		330 + R _n ; I = 1	[(ADR) ((ADR)) 377 _g + ((ADR)) * 2 ⁸] ∧ MASK

d) traitement du bit

CPZRO	ADR, CONF	77	(ADR) CONF → α saut à RAC + 1 si α = 0
CPONE	ADR, CONF	73	(ADR) ∧ CONF → α saut si α = CONF
SETONE	ADR, CONF	74	(ADR) CONF → ADR
SETZRO	ADR, CONF	70	(ADR) CONF → ADR
e) traitement du mot			
SHLA, NBRE		373	A _L → A _H ; 0 → A ₀
SHLB, NBRE		370	B _L → B _H ; 0 → B ₀
SHRA, NBRE		371	A _H → A _L ; 0 → A ₂₄
SHRB, NBRE		372	B _H → B _L ; 0 → B ₂₄

RORA, NBRE	375	A _H → A _L ; A ₀ → A ₂₄
RORB, NBRE	374	B _H → B _L ; B ₀ → B ₂₄
ROLA, NBRE	377	A _L → A _H ; A ₂₄ → A ₀
ROLB, NBRE	376	B _L → B _H ; B ₂₄ → B ₀

3) groupe arithmétique | logique à référence interne

RADD, R _j	360 + R _j	j = A A + B → A j = B A + B → B
RSUB, R _j	362 + R _j	j = A A - B → A j = B B - A → B
CP, R _n	364 + R _n	(R _n) → R _n
IRJ, R _n ADR	21 + 4 * R _n	(R _n) + 1 → R _n ; saut à ADR si (R _n) ≠ 0
DRJ, R _n ADR	22 + 4 * R _n	(R _n) - 1 → R _n ; saut à ADR si (R _n) = 0
TZR, R _n	123 + 4 * R _n	0 → R _n
RAND, R _n	120 + 4 * R _n	A R _n → A
ROR, R _n	121 + 4 * R _n	A ⊕ R _n → A
COPY, SR _n , D _{r_n} DATA	340 + SR _n * 4 + DR _n	(SR _n) + DATA → DR _n

4) groupe des instructions de contrôle

JMP	ADR	42	Saut à ADR
JMP	⊙ ADR	42 ; I = 1	Saut à (ADR)
JMPI, R _n	ADR	2 + 4 * R _n	Saut à ADR + (R _n)
JMPI, R _n ⊙ ADR		2 + 4 * R _n ; I = 1	Saut à (ADR + (R _n))
JZRO, R _n ADR		20 + 4 * R _n	Saut si R _n = 0
JZRO, R _n ⊙ ADR		20 + 4 * R _n ; I = 1	Saut si R _n < 0
JNZR, R _n ADR		23 + 4 * R _n	Saut si R _n < 0
JNZR, R _n ⊙ ADR		23 + 4 * R _n ; I = 1	Saut si carry = 0
JTC	ADR	50	Saut si carry = 0
JTC	⊙ ADR	50 ; I = 1	Saut avec marquage
JMPM	ADR	63	Saut avec marquage
JMPM	⊙ ADR	63 ; I = 1	Saut avec marquage

5) groupe des entrées/sorties

OUTD ORDRE, QOX1, QOX1, Q1X0	072	exécution d'un ordre CAMAC mémorise en ORDRE avec débranchement selon valeur de Q et de X
IND ORDRE, QOX0, QOX1, Q1X0	071	idem
IUBC ORDRE, ADIN, QX	250	transfert de bloc CAMAC en REPEAT MODE (A) = Nombre de mots
IMAD ORDRI, ORDRF, NEMOTS, ADIN		transfert de bloc CAMAC en AUTO-SCAN
STDMA, R _n ADR	1 + 4 * R _n	lancement DMA
STDMA, R _n @ ADR	1 + 4 * R _n ; I = 1	
ENDMA, R _n ADR	252	fin de DMA
ENDMA, R _n @ ADR	252 ; I = 1	

6) groupe : Instructions "système"

ION	140	valider les interruptions
IOFF	141	invalider les interruptions
MSKAL	142	masquer tous les niveaux
CLAL	143	désactiver tous les niveaux
MSKLV NIV	144	masque le niveau NIV
DSKLV NIV	145	démarquer le niveau NIV
CLLV NIV	146	désactiver le niveau NIV
ACTLV NIV	147	activer le niveau NIV
RCLV	166, 4000	lire niveau en cours
GIVUP	167, 4000	quitter le niveau courant

7) Groupe : multiplieur

FLOAD, R _n ADR	260 + R _n	(R _n) _{FL} + (ADR) _{FL} → (R _n) _{FL} saut à RAC + 1 si OVERFLOW
FLOMUL, R _j ADR	270 + R _j	(R _j) _{FL} * (ADR) _{FL} → (R _j) _{FL} saut à RAC + 1 si OVERFLOW
INTMUL	275	(A) _E x (B) _E → (A.B) _E
FLOINT ADR	276	(A) _{FL} → (A) _E saut à ADR si OVERFLOW
INTFLO	277	(A) _E → (A) _{FL}

REFERENCES

- 1) S.D.DRELL and T.M.YAN, Ann.of Phys.66, 578 (1971)
- 2) ANDERSON et al., "Production of Muon pairs by 225 GeV π^+ , K^+ , p, \bar{p} beams on Nuclear Targets" - Preprint EF 179-6, 79-7, 79-8
- 3) F.M.RENARD, "Aspects théoriques de la production de leptons dans les collisions hadroniques", Gif 77
- 4) G.ALTAPELLI, "Partons in quantum chromodynamics" Gif 77
- 5) P.SCHUBELIN, "Drell-Yan dileptons and associated hadrons", XIIème International School on High Energy Physics - Primorsko, Bulgarie (1978), CRN/HE 79-3.
- 6) J.CLEYMANS and M.KURODA, "Angular distribution of dileptons in hadronic collisions", preprint BI-TP 78/28 - Déc. 1978
- 7) F.VANUCCI, "High Energy muon-pair experiments and the Drell-Yan model", International School of Elementary Particle Physics - Kupari - Dubrovnik, Yougoslavie, Sept. 1979, CERN-EP/79-151
- 8) J.K.YOH et al., Phys.Rev.Lett.41, 684 (1978) ; 41, 1083 (1978).
- 9) D.M.KAPLAN et al., Phys.Rev.Lett.40, 435 (1978).
- 10) L.M.LEDERMAN, Proc. 19th Int. Conf. on High-Energy Physics, Tokyo, 1978 (Phys.Soc.Japan, Tokyo, 1979), p.706.
- 11) C.B.NEWMAN et al., Phys.Rev.Lett.42, 951 (1979)
- 12) G.E.HOGAN et al., Phys.Rev.Lett.42, 948 (1979)
- 13) K.J.ANDERSON et al., Phys.Rev.Lett.42, 944 (1979)
- 14) J.BADIER et al., preprints CERN-EP/79-61 (1979) and CERN-EP/79-68 (1979).
- 15) Contribution à l'étude des systèmes informatiques en temps réels en physique des particules élémentaires - Thèse de doctorat ès sciences J.M.MEYER (1977) Université du Haut-Rhin
- 16) Contribution à l'étude et à la réalisation d'un processeur microprogrammable, ultra-rapide, destiné à la gestion des tâches CAMAC - thèse de doctorat de 3ème cycle - C.BOUILIN (1976) Université Louis Pasteur Strasbourg I
- 17) Conception et réalisation d'un système de développement exécutable sur mini-ordinateur NORD-10, pour un microprocesseur spécialisé - thèse de doctorat de 3è cycle - G.HUOT (1978) Université Louis Pasteur Strasbourg I

- 14) La microprogrammation - G.BOULAYE, Dunod (1972)
- 15) NORD-10 Référencemanuel - SINTRAN III Users Guide NORSK DATA ELEKTRONIKK
- 16) 3001 Microprogram Control Unit - Document Intel MCS 149-774-1K
- 17) 3002 Control Processing Element - Document Intel MCS 269-0275/27-5K
- 18) Multiprocessor acquisition system including a special purpose CAMAC dedicated microprocessor - J.LECOQ, M.PERRIN, J.M.MEYER, C.BOUILIN, G.METZGER (CRN-HE 78-16) - Contribution to CAMAC Symposium, September 25-29, 1978, Hambourg
- 19) Mémoires mortes ou vives : comment les tester ? C.BOUCHET - Electronique et Applications industrielles - 1980 - 280
- 20) Séquences de test pour mémoires à semi conducteurs - J.MONTOIS - Electronique et Applications industrielles - 1975 - 206

REMERCIEMENTS

Je prie Monsieur M.CROISSIAUX de bien vouloir trouver ici l'expression de ma profonde gratitude pour l'accueil qu'il m'a réservé à la Division des Hautes Energies, et pour les conseils et les encouragements qu'il n'a cessé de me prodiguer.

Je remercie Monsieur le Professeur G.MONSONEGO de m'avoir fait l'honneur d'accepter la présidence du jury de thèse.

Monsieur J.M.MEYER a accepté de corriger cette thèse et de faire partie du jury : qu'il soit ici remercié.

Ce travail a été guidé par J.J.BLAISING pour la partie physique et par J.LECOQ pour la partie microinformatique et les tests : ils m'ont permis d'acquérir les compétences nécessaires à la réalisation de cette tâche ; ils m'ont aidé sans compter, je les en remercie profondément.

Mes remerciements vont également à J.D.BERST, qui dans les moments difficiles joua souvent le rôle de la bonne fée.

Je tiens également à remercier tous les stagiaires, en particulier P.KLEIN et J.M.SCHEIBER, qui ont assumé une grande partie de la tâche ingrate du câblage et des tests.

Mme M.GOETZ a assuré avec diligence la dactylographie de cette thèse : qu'elle soit ici remerciée.